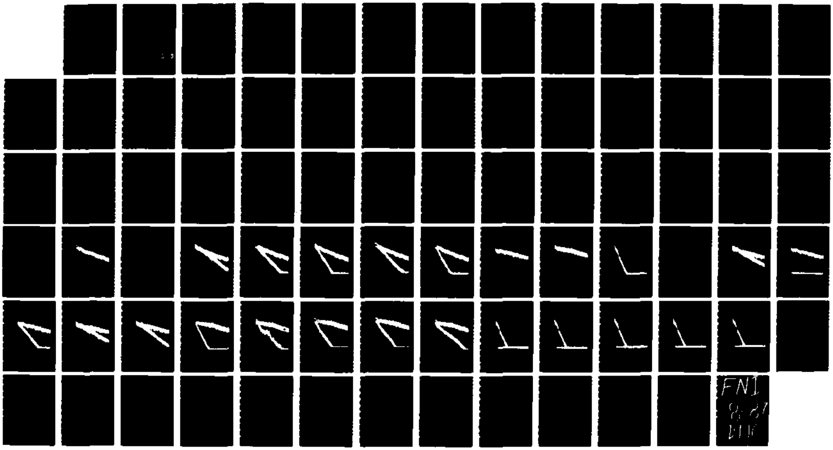
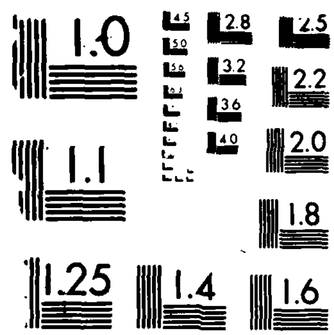
AD-A182 829 A COMPARISON OF TRANSFORM DOMAIN ADAPTIVE FILTERS WITH EMPHASIS ON THE HA (U) ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB J J MURPHY JUN 87 1/1

UNCLASSIFIED UILU-ENG-87-2234 N00014-84-C-0149 F/G 12/2 NL





MICROCOPY RESOLUTION TEST CHART

June 1987 UILU-ENG-87-2234

COORDINATED SCIENCE LABORATORY
College of Engineering

A COMPARISON OF TRANSFORM DOMAIN ADAPTIVE FILTERS, WITH EMPHASIS ON THE HARTLEY TRANSFORM

James Joseph Murphy

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

A182 829

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-87-2234	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research (for JSEP) AT&T
6c. ADDRESS (City, State, and ZIP Code) 1101 W. Springfield Avenue Urbana, IL 61801	7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy St. Arlington, VA 22217 (ONR) AT&T Information Systems Middleton, NJ
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Joint Services Electronics Program and AT&T 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-84-C-0149 (JSEP)
8c. ADDRESS (City, State, and ZIP Code) Office of Naval Research 800 N. Quincy St. Arlington, VA 22217 AT&T Information Systems Middleton, NJ	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
A Comparison of Transform Domain Adaptive Filters, with Emphasis on the Hartley Transform

12. PERSONAL AUTHOR(S)
Murphy, James Joseph

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	June 1987	77

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	adaptive filters, orthogonal transforms

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The least-mean-square (LMS) algorithm is the most often used real-time adaptive filtering algorithm due to its computational simplicity and remarkably good fit to the optimal Wiener solution. There have been many transform domain algorithms proposed for improving the convergence rate of the LMS algorithm, the most popular of which has been the Discrete Fourier Transform (DFT). However, the DFT requires complex arithmetic and, thus, has proven computationally undesirable for applications involving only real signals. A number of unitary, real transforms have been proposed as less costly replacements for the DFT. These include the Discrete Cosine Transform (DCT), the Discrete Walsh-Hadamard Transform (WHT), and the Power-of-Two Transform (PO2). Each of these in some way exhibits a property necessary to speed the convergence rate, at a lower computational cost than the DFT. This work investigates the use of another real transform, the Discrete Hartley transform (DHT), for adaptive system estimation, and adaptive echo cancelling. It is shown that the DHT performs better than these other real transforms under most circumstances. Its relationship to the DFT is such that it can be transformed into the DFT with simple algebraic manipulation.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

A COMPARISON OF TRANSFORM DOMAIN ADAPTIVE FILTERS, WITH EMPHASIS ON THE HARTLEY TRANSFORM

BY

JAMES JOSEPH MURPHY

B.Eng., Manhattan College, 1985

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1987

Urbana, Illinois

ABSTRACT

The least-mean-square(LMS) algorithm is the most often used real-time adaptive filtering algorithm due to its computational simplicity and remarkably good fit to the optimal Wiener solution. There have been many transform domain algorithms proposed for improving the convergence rate of the LMS algorithm, the most popular of which has been the Discrete Fourier Transform, (DFT). However, the DFT requires complex arithmetic, and thus has proven computationally undesirable for applications involving only real signals. A number of unitary, real transforms have been proposed as less costly replacements for the DFT. These include the Discrete Cosine Transform (DCT), the Discrete Walsh-Hadamard Transform (WHT), and the Power-of-Two Transform (PO2). Each of these in some way exhibits a property necessary to speed the convergence rate, at a lower computational cost than the DFT. This work investigates the use of another real transform, the Discrete Hartley Transform, (DHT), for adaptive system estimation, and adaptive echo cancelling. It is shown that the DHT performs better than these other real transforms under most circumstances. Its relationship to the DFT is such that it can be transformed into the DFT with simple algebraic manipulation.

DEDICATION

This work is dedicated to James J. Murphy, Sr., the first engineer in the family.

ACKNOWLEDGEMENT

I would like to thank my advisor, Professor W. Kenneth Jenkins, without whose patient assistance this work would not have been possible.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Lately, research has been conducted on the use of orthogonal transforms to speed the convergence rate of adaptive systems. Work has been done to examine the usefulness of certain well-known transforms for this purpose [7,8], and new orthogonal transforms have been designed to operate effectively under a given set of circumstances [9]. When the system is required to adapt under many different signalling conditions, it becomes difficult to select a transform that works well under all circumstances. There is a need to compare many of these established transforms to one another and to rank them according to their ability to provide improved performance in adaptive filters. This research provides such a comparison for most of the known transforms used for adaptive filtering the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Walsh-Hadamard Transform (WHT). It also compares the less widely known Power-of-Two Transform (PO2) and introduces the Discrete Hartley Transform (DHT) to the field. The Hartley Transform will be shown to improve adaptive behavior more consistently than the other transforms, at an equal or lower computational cost.

1.1 Past Work, and Intent

A considerable amount of the preliminary work in transform domain adaptive

filtering studies was done here at the University of Illinois by D. F. Marshall, and J. R. Kreidle [7,9,17]. Their work was based on a combination of the LMS algorithm developed by Widrow and Hoff [2] and the transform domain LMS algorithm developed by Narayan et al. [8]. The objective of this research is to improve upon their results and to present a hierarchical ordering of transform performance in FIR adaptive filters, as determined from a series of computer experiments in which these transforms were used to speed the convergence rate of the adaptive LMS algorithm. Considered in these experiments are the Discrete Fourier, Discrete Cosine, Discrete Walsh-Hadamard, and Power-of-Two Transforms. Several authors have shown the usefulness of these suboptimal transforms for various inputs [7,8,9]. This research will attempt to rank these transforms and will, in addition, introduce another suboptimal transform, the Discrete Hartley Transform, that will be shown to perform better under most conditions than any of the above.

It is hoped that the transform domain adaptive filtering algorithms discussed in this work will find practical application in many telecommunications problems. One possible application occurs in a telephone channel carrying voice or data signals with the need for echo cancelling. The channel will be modelled as a low-pass system, which the adaptive filter will try to match when presented with various colored noise inputs. It is important that the echo canceller converge in the presence of many different signalling conditions, since modern telecommunications networks are used for more than simple voice transmission, and these other signals (data, tones, etc.) sometimes make convergence difficult in the time domain. The situation is improved by using the transformed inputs,

but results are not uniform with different transforms. A second criterion is that the method used to speed convergence must fit within the practical constraints of modern VLSI technology for size and power consumption. The interest in the real transforms stems from this stipulation, because the FFT algorithms that are known require more space and power than some situations will allow. While this is the application in mind throughout these studies, it is not claimed that the model used accurately represents a channel of a telephone system; it is merely an easily controllable test case that serves to study the performance of these transform domain algorithms.

CHAPTER 2

TRANSFORM DOMAIN ADAPTIVE FILTERING

2.1 Background: The LMS Algorithm

The tapped-delay-line FIR adaptive filter that forms the basis of this work is shown in Figure 2.1. The input signal vector is defined as

$$X(n)=[x(n)\ x(n-1)\ \cdots\ x(n-N-1)]'$$

and the weight vector is

$$A(n)=[a(n_0)\ a(n_1)\ \cdots\ a(nN-1)]'$$

where ' denotes transpose. The input X(n) is assumed to be a sequence of zero-mean, stationary random variables which are not necessarily uncorrelated. The filter output is a scalar y(n) given by

$$y(n)=X(n)'A(n).$$

To generate the error needed for adaptation, the desired response d(n), a scalar, is compared to the output y(n) to give $\varepsilon(n)=d(n)-y(n)$. The error is used by the LMS

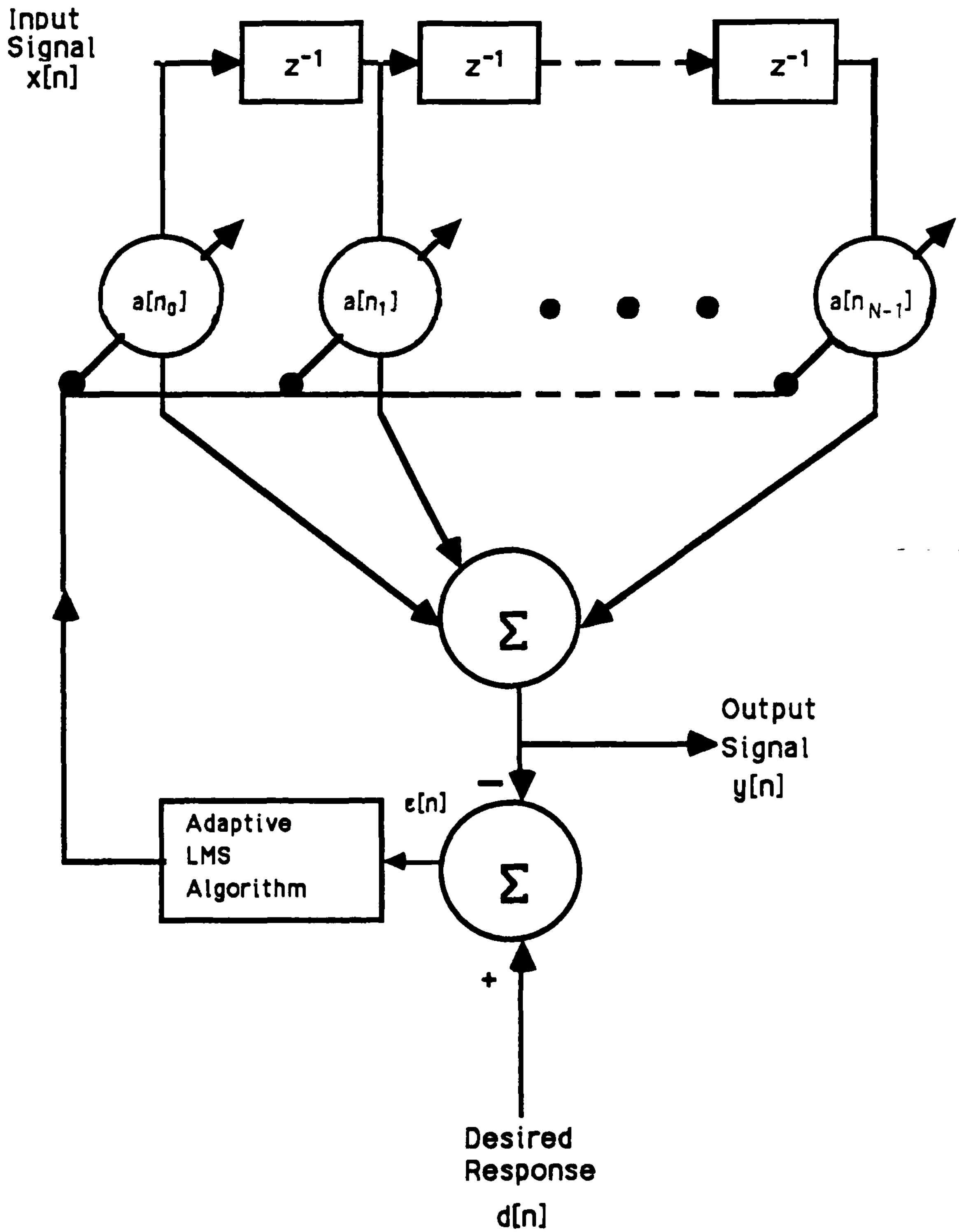


Figure 2.1

The tapped delay line adaptive filter

algorithm to update the weight vector A(n) with each sample according to the expression

$$A(n+1)=A(n)+2\cdot\mu\cdot X(n)\cdot\varepsilon(n)$$

where μ is a constant called the step size which is usually determined experimentally.

At convergence, when μ is properly chosen, the expected value of the weight vector $A(\infty)$ is the Wiener solution

$$E[A(\infty)]=R_{xx}(i,j)^{-1}R_{xd}(i,j) \quad i,j=0,1,\ldots,N-1.$$

The Wiener solution describes the optimal solution (solution of least mean square error). The main attraction of the LMS algorithm is its simplicity, which comes from the fact that it uses an estimate of the gradient rather than the true gradient. Any misadjustment of the solution can be attributed to this gradient estimate.

It can be seen that the LMS algorithm converges very quickly if the input is uncorrelated, stationary, and the unknown system is a linear FIR. If we relax one or more of the conditions, however, the algorithm will not converge as rapidly to the Wiener solution. In particular, if the stipulation of uncorrelated input signal is relaxed, as will happen under all practical signalling conditions, transform domain techniques appear to be useful to retain the simplicity of the LMS algorithm, while elevating the performance to the level of more complex methods.

To process the input data in a transform domain it must first be multiplied by a unitary matrix W, the transform matrix, which contains orthogonal rows and columns to form a new input vector $Z(n)=W \cdot X(n)$. Figure 2.2 shows the adaptive filter with the transform matrix in place.

2.2 Interpretation

The desired effect of transforming the input vector is to partially "whiten" the signal in an attempt to increase the efficiency of the LMS algorithm. A "white" signal, also called a decorrelated signal, is one whose correlation matrix R_{XX} is diagonal, with all diagonal elements equal to one. The transform brings about this whitening by partially diagonalizing the input correlation matrix, and then equalizing the values along the diagonal (the eigenvalues of the matrix) to some extent by normalizing them with a power factor $\sigma_i^2=E[|x_i(n)|^2]$ (Equation (2.1)) [17]. An attempt has been made to model the orthogonalization process as the action of a bank of band-pass filters, with each orthogonal basis function representing the center frequency of the associated filter [7]. Thus for an eight tap system, as was used in the following simulations, there are eight equally spaced bands around the normalized frequency axis. Each filter has associated with it one point at which the input is completely decorrelated, and many points at which varying amounts of correlation remain. It is this partial nature of the decorrelation that makes transform techniques less than ideal. Only one transform, the Karhunen-Loeve, is able to completely decorrelate any input. Other transforms succeed to varying degrees. An attempt was made to describe this

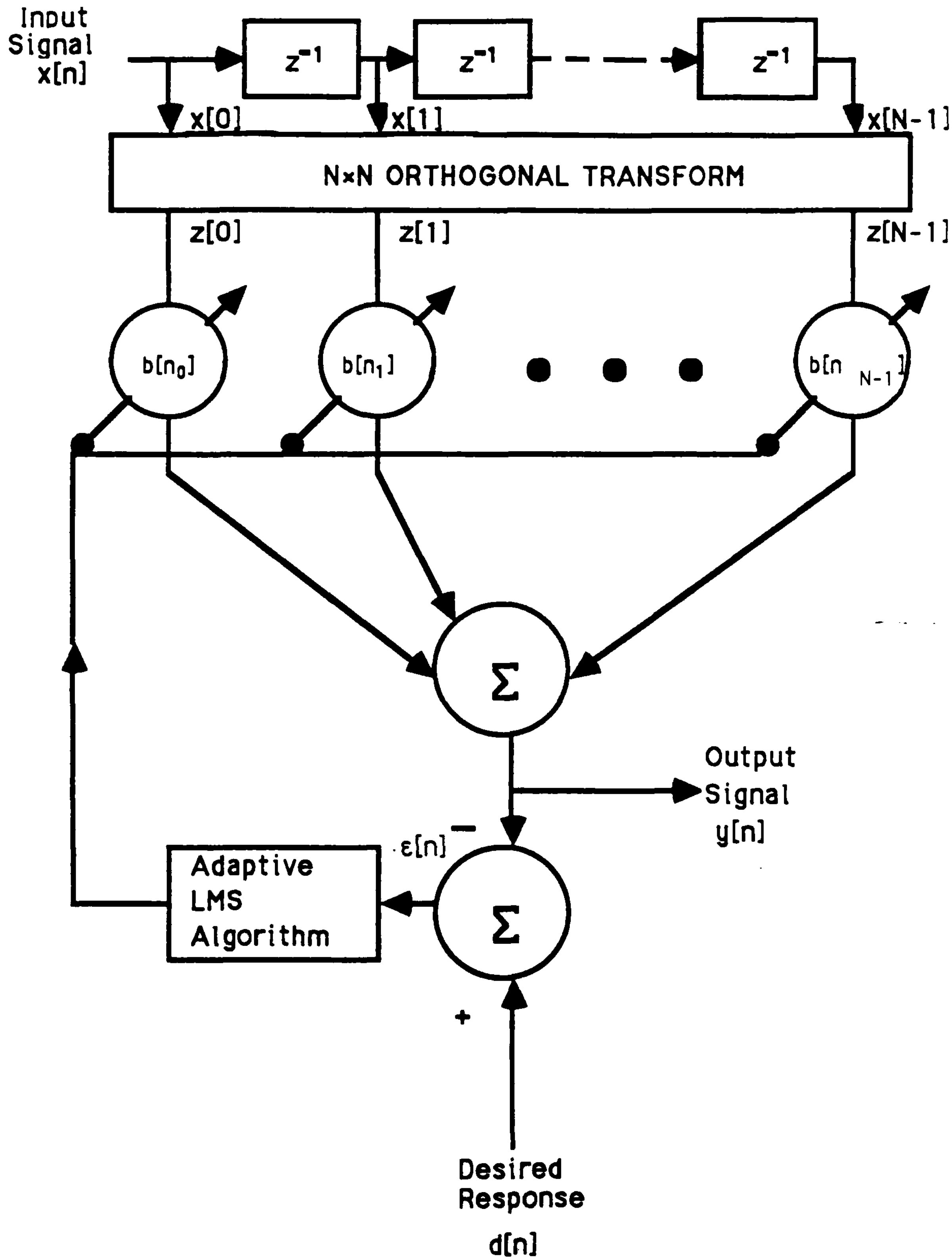


Figure 2.2

An FIR adaptive filter using an orthogonal transform to improve convergence

decorrelation, and to find a way to measure a transform's effectiveness. The literature was surveyed, and a measure was found for the decorrelating ability of a unitary transform for a class of inputs described as Markov-1 processes. A Markov-1 process is an autoregressive process whose statistics are first order. The decorrelation measure was well defined for this type of process, but was not defined for any other order of input statistics. Perhaps this is so because the first order processes are much simpler than any other. An attempt was made to use the decorrelation measure for other types of processes. The measure was found to be invalid for processes other than first order Markov. A further attempt was made to create a similar measure for other processes. This was also unsuccessful. The analysis was left at this point in favor of an experimental study of transforms' effectiveness that comprises the body of this work. The design of a useful measure of decorrelation remains a valid area of research for the future [18,19].

2.3 The Karhunen-Loeve Transform

An optimal orthogonal transform for adaptive filtering must be one that will, for any input vector, result in completely uncorrelated outputs, with equal power in each sample (in other words, white noise). This is necessary because the LMS algorithm will only converge quickly if the input is in this form. The only transform capable of completely decorrelating any input vector is the well known Karhunen-Loève Transform. The transform matrix contains elements which are functions of the input autocorrelation matrix, so some specific knowledge of

the input is needed to implement this method. The problem is it is very difficult to update these input statistics on line in real time. An estimate of the autocorrelation can be made as the data are input to the system, but this will add another time-varying filter. Thus, it is of interest to find suboptimal transforms with fixed statistics that can perform reasonably well on a wide variety of input signals.

CHAPTER 3

THE TRANSFORMS

3.1 A Complex Transform: The Discrete Fourier Transform

The DFT is a familiar example of an orthogonal transform. If no specific type of input is assumed, then the DFT decomposes the input into components which are not completely uncorrelated [10]. If, on the other hand, the input is periodic, the Fourier coefficients will be uncorrelated. For the DFT, the disjoint filter interpretation provides a useful insight. For an eight-point DFT, there are eight points in the Fourier domain where there is zero overlap of the band-pass filters that are spaced equally around the unit circle in the z-plane. This means there are only eight frequencies where the process is truly decorrelated.

If the input was periodic with frequency content only at these eight points, the DFT would produce an uncorrelated output. The autocorrelation matrix of a periodic input is given by the diagonal matrix

$$R_{zz}(i,j) = \text{diag} \left| r_{zz}(i,i) \right| ; \quad i=0,\ldots,N-1 \tag{3.1}$$

where the r_{zz} are the diagonal elements of the matrix.

The average power in a sequence is defined by $\sigma_z^2 = r_{zz}(i,i)$. Therefore, if the eight components in the input are of different average powers, then Equation

(3.1) is a diagonal matrix with unequal elements. To finish the "whitening" of this case, it is simply necessary to multiply each diagonal element by the inverse of the average power. This is the power normalization of Equation (2.1).

The above is a very special case, where the input has only N frequency values for an N-point transform. For other inputs the degree of decorrelation can be estimated to a degree by using the disjoint filter interpretation [7]. The decorrelation in the general case will be incomplete, so a system comprised of a partially decorrelated input should not be expected to converge as quickly as one whose input is white. However, the transformed input should converge more quickly than the time domain algorithm with the same input.

3.2 The Real Transforms

Real orthogonal transforms differ from the DFT by the fact that they require only real multiplications and additions to calculate an output from a real input, while the DFT will require complex mathematics even with a real input. The four real transforms that are used are the Discrete Cosine Transform (DCT), Discrete Hartley Transform (DHT), Discrete Walsh-Hadamard Transform (WHT), and Power-of-Two Transform (PO2).

3.2.1 The discrete cosine transform

The DCT is defined by

$$F(k) = \frac{2c(k)}{N} \sum_{N-1}^{n=0} f(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right]$$

where k = 0,....,N-1 and

$$c(k) = \begin{bmatrix} \frac{1}{\sqrt{2}}, k = 0 \\ 1, k = 1,....,N-1 \\ 0, k \text{ elsewhere} \end{bmatrix}.$$

A major disadvantage to the DCT is that it does not have an FFT-like fast algorithm, although fast algorithms have been derived for a large set of specific inputs [11]. So for each value of N, the fast algorithm for the DCT changes, rather than being an extension of a general form. This can make the transform less desirable under the most general of circumstances, that is, when the input is not only of unknown type, but the order of the system is not fixed either.

3.2.2 The discrete Walsh-Hadamard

From the point of computation, the WHT is the most desirable. The basis functions consist of sampled Walsh functions, which are an orthogonal set of piecewise constant binary valued functions. The first six Walsh functions, and

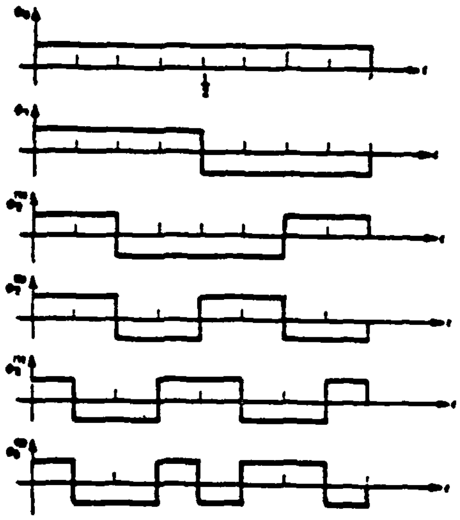


Figure 3.1a First six Walsh functions

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Figure 3.1b The matrix for the Walsh Hadamard Transform (N=8)

the matrix for the transform with N=8 are shown in Figures 3.1a and 3.1b respectively.

Even if the WHT were not a strongly performing transform, the simplicity of its calculation would make it appealing in many applications where computation is more critical, and only some convergence improvement is needed over the time domain. Calculation of the fast algorithm requires no multiplications, as all matrix elements are ±1, and only $N\log_2 N$ additions. The fast algorithm has the same form as the FFT, and can in fact be calculated from the FFT by making all variables real, and setting all the complex exponentials equal to ±1.

3.2.3 Power-of-two transform

The PO2 is a transform custom-designed at the University of Illinois to be useful in transform adaptive digital filtering, while keeping a simple computational form. It is called the Power-of-Two Transform because the elements of the transform matrix were required not only to be real, but also to be simple powers-of-2. This implies that in practice the PO2 can be implemented entirely without multiplication, i.e., it requires only shifting and adding [9].

The transform matrix for the PO2 with N=8 is shown in Figure 3.2. With this length transform, only four power-of-2 levels are used. It is thought that for longer transform lengths, more distinct levels of powers would appear [10].

$$
\begin{bmatrix}
1.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 1.0 \\
2.0 & -1.0 & 2.0 & -2.0 & 2.0 & -2.0 & 1.0 & -2.0 \\
2.0 & -2.0 & -1.0 & 2.0 & -2.0 & 1.0 & 2.0 & -2.0 \\
2.0 & 2.0 & -2.0 & -1.0 & 1.0 & 2.0 & -2.0 & -2.0 \\
2.0 & -2.0 & 2.0 & -1.0 & -1.0 & 2.0 & -2.0 & 2.0 \\
2.0 & 2.0 & -1.0 & -2.0 & -2.0 & -1.0 & 2.0 & 2.0 \\
2.0 & -1.0 & -2.0 & 2.0 & 2.0 & -2.0 & -1.0 & 2.0 \\
1.0 & 2.0 & 2.0 & 2.0 & -2.0 & -2.0 & -2.0 & -1.0
\end{bmatrix}
$$

Figure 3.2 A specially designed Power-of-Two Transform (N=8)

CHAPTER 4

THE HARTLEY TRANSFORM

4.1 The Continuous Time Hartley Transform

The Hartley Transform is a real integral transform named for Ralph V. Hartley, who formulated it and published the results of his findings in the Proceedings of the Institute of Radio Engineers [12], in 1942. Given a time function V(t), a transform pair $V(t) <-> \psi(\omega)$ is defined by Equations (4.1) and (4.2) as

$$\psi(\omega)= 1/\sqrt{2\pi} \int_{-\infty}^{\infty} V(t) \operatorname{cas} \omega t \, dt \tag{4.1}$$

$$V(t)= 1/\sqrt{2\pi} \int_{-\infty}^{\infty} \psi(\omega) \operatorname{cas} \omega t \, d\omega \tag{4.2}$$

where

$$\operatorname{cas} t \equiv \cos t + \sin t .$$

It is seen that the transform and its reciprocal have exactly the same form, and since V(t) is real, the transform is also real. Furthermore, if we write the Fourier Transform as

$$S(\omega)= 1/\sqrt{2\pi} \int_{-\infty}^{\infty} V(t) \exp(-i\omega t) \, dt$$

and its inverse as

$$V(t) = 1/\sqrt{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp(i\omega t)\, d\omega,$$

we can convert the Hartley Transform to the Fourier by a simple algebraic manipulation [4]. Let $\psi(\omega) = e(\omega) + o(\omega)$, where $e(\omega)$ and $o(\omega)$ are the even and odd parts of $\psi(\omega)$, respectively. Then

$$e(\omega) = [\psi(\omega) + \psi(-\omega)]/2 = 1/\sqrt{2\pi} \int_{-\infty}^{\infty} V(t) \cos \omega t \, dt$$

and

$$o(\omega) = [\psi(\omega) - \psi(-\omega)]/2 = 1/\sqrt{2\pi} \int_{-\infty}^{\infty} V(t) \sin \omega t \, dt.$$

So given $\psi(\omega)$,

$$S(\omega) = e(\omega) - io(\omega).$$

Conversely, given $S(\omega)$,

$$\psi(\omega) = \mathrm{Re}[S(\omega)] - \mathrm{Im}[S(\omega)].$$

This shows that the Hartley Transform allows the calculation of the Fourier Transform without the use of complex arithmetic, resulting in savings both in computation and storage in situations where the Fourier Transform is desired.

In the case of the Hartley Transform, there are a number of known properties that can streamline calculations. These are expressed as theorems like those of the Fourier Transform, and in most cases have been developed from a corresponding Fourier theorem. They are described in detail in the literature [4].

4.2 The Discrete Hartley Transform

For the Hartley Transform to be of use in the problem of adaptive filtering, there must be a discrete Hartley Transform, and an FFT-like fast algorithm accompanying the continuous case. Fortunately both of these have been worked out in some detail [3,4,5,6].

If we substitute for time a discrete variable τ, which can take on N integral values from 0 to N-1, we can define the Discrete Hartley Transform (DHT) as

$$H(\upsilon) = N\text{-}1 \sum_{\tau=0}^{N-1} f(\tau)\, \mathrm{cas}(2\pi\upsilon\tau/N)$$

and its inverse as

$$f(\tau) = \sum_{\upsilon=0}^{N-1} H(\upsilon)\, \mathrm{cas}(2\pi\upsilon\tau/N)$$

where cas(x) was defined above. Note that in the discrete case, as with the continuous Hartley Transform, the transform and its inverse have the exact same form. Similarly, we expect a simple algebraic relationship between the DHT and the DFT. To get the DFT, we again split the DHT into even and odd parts

$$H(\upsilon) = E(\upsilon) + O(\upsilon)$$

where

$$E(\upsilon) = [H(\upsilon) + H(N-\upsilon)]/2$$

and

$$O(\upsilon) = [H(\upsilon) - H(N-\upsilon)]/2.$$

Then the DFT is

$$F(\upsilon) = E(\upsilon) - \iota O(\upsilon).$$

Once again, conversely, the DHT is

$$H(\upsilon) = \mathrm{Re}[F(\upsilon)] - \mathrm{Im}[F(\upsilon)].$$

Again, in the discrete case, as in the continuous time case, there are theorems for the DHT that the interested reader may wish to examine [3,4]. Figure 4.1 shows the form of the DHT for N=8. Notice the elements of the matrix are either ±1, 0, or $\pm\sqrt{2}$. This reduces the number of multiplications that must be done, much like the PO2 or WHT require.

4.3 The Fast Hartley Transform

It was discussed earlier that for the DHT to be really useful, it must have a fast algorithm. Since it resembles the DFT in so many ways already, one could surmise that it would have one. A Fast Hartley Transform has been developed that, given a data sequence of length N, where N = 2p, will reduce the number of arithmetic operations to the order of $N\log_2 N$ [6].

The transform takes on a butterfly form, resembling the forms of the FFT, and is shown in Figure 4.2. What follows is a brief description of each step of

$$
\frac{1}{8}\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \sqrt{2} & 1 & 0 & -1 & -\sqrt{2} & -1 & -0 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 0 & -1 & \sqrt{2} & -1 & 0 & 1 & -\sqrt{2} \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -\sqrt{2} & 1 & 0 & -1 & \sqrt{2} & -1 & 0 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 0 & -1 & -\sqrt{2} & -1 & 0 & 1 & \sqrt{2}
\end{bmatrix}
$$

Figure 4.2 The discrete Hartley Transform (N=8)

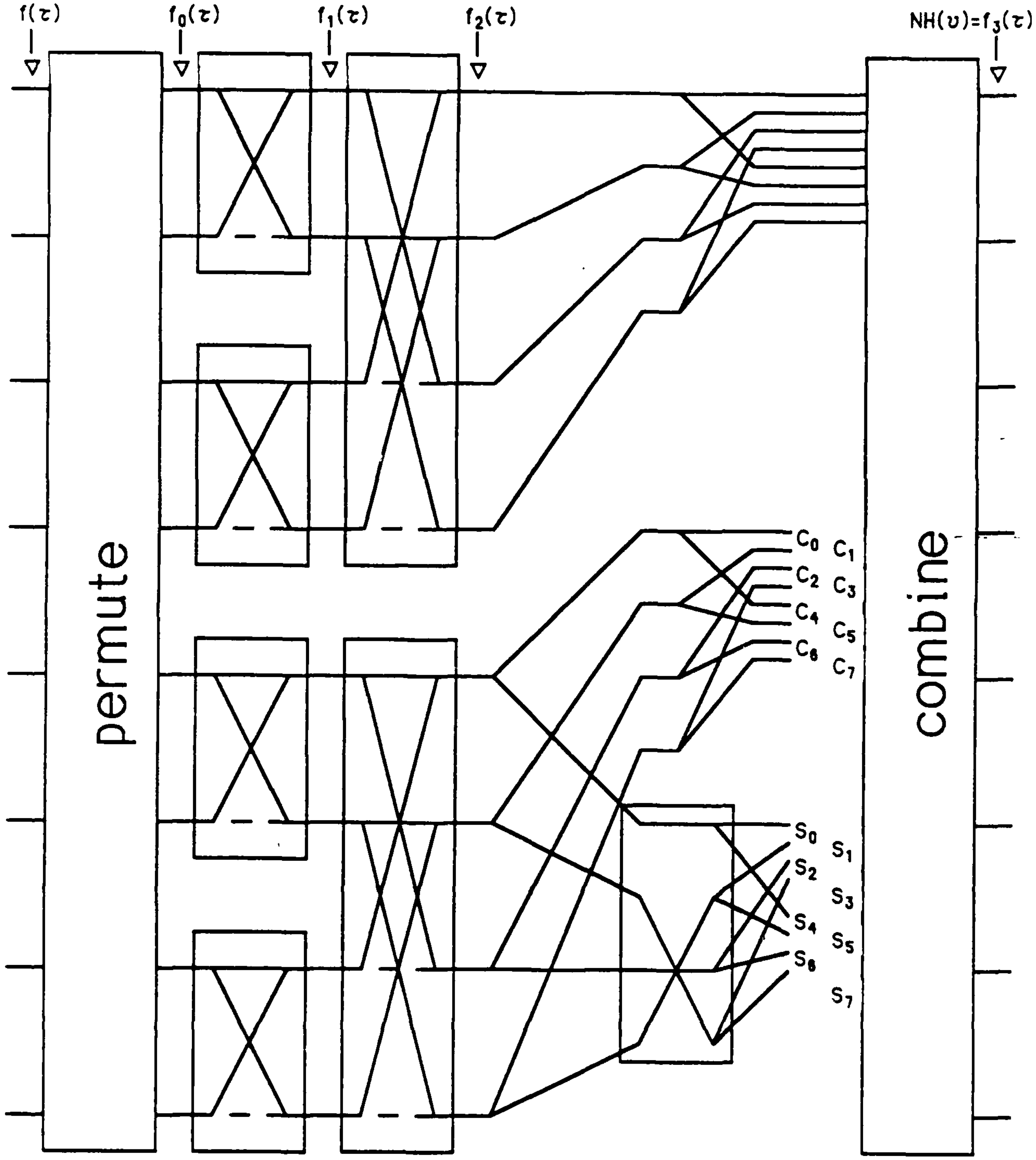


Figure 4.2

Fast Hartley Transform flow diagram

the process. The purpose of PERMUTE is to arrange the input vector in data pairs. So an eight-element input vector would be arranged as four pairs, etc.. The output $F_0(\tau)$ is the result of the first "butterfly", which furnishes the two elements to be combined for the result called $f_1(\tau)$. This process continues a total of P-1 times. The COMBINE operation performs a function like the bit reversal required for the FFT. The result $f_{p-1}(\tau)$ will be equal to $N \times H(\upsilon)$. Dividing each value of $f_{p-1}(\upsilon)$ by N will give the correct value of $H(\upsilon)$. We may also calculate the DFT at this point, if so desired, via the method described earlier.

CHAPTER 5

COMPUTER EXPERIMENTS

5.1 Computer Methods

All experiments were run on the Cyber-175 in FORTRAN. It was necessary to generate a model system, input vector, coloring noise filter, and the adaptive LMS algorithm for each transform. An eight-tap FIR low-pass filter system was created with DFDZR1 [14], a FORTRAN version of the Parks-McClellan equiripple filter design program. The frequency response of the resulting filter, shown in Figures 5.1a and 5.1b, was then used as the model system for the algorithm. White noise was generated by RANF, an intrinsic function on CYBER. Because RANF results in random samples between 0.0 and 1.0., 0.5 was subtracted from each value of RANF in order to have a random variable with zero-mean to be used as the input vector. Four coloring noise filters, each also an eight-tap FIR filter, were generated to test the tranforms robustness. The program to actually perform the simulations was ADAPOP4 [13]. Other software was created to plot the resulting data.

5.2 Experiments and Results

It was hoped that by simulating the LMS algorithm using a different

$$h_d(n) = \begin{bmatrix} -.033 \\ -.089 \\ .099 \\ .478 \\ .478 \\ .099 \\ -.089 \\ -.033 \end{bmatrix}$$

Figure 5.1a Impulse response of eight-tap unknown filter

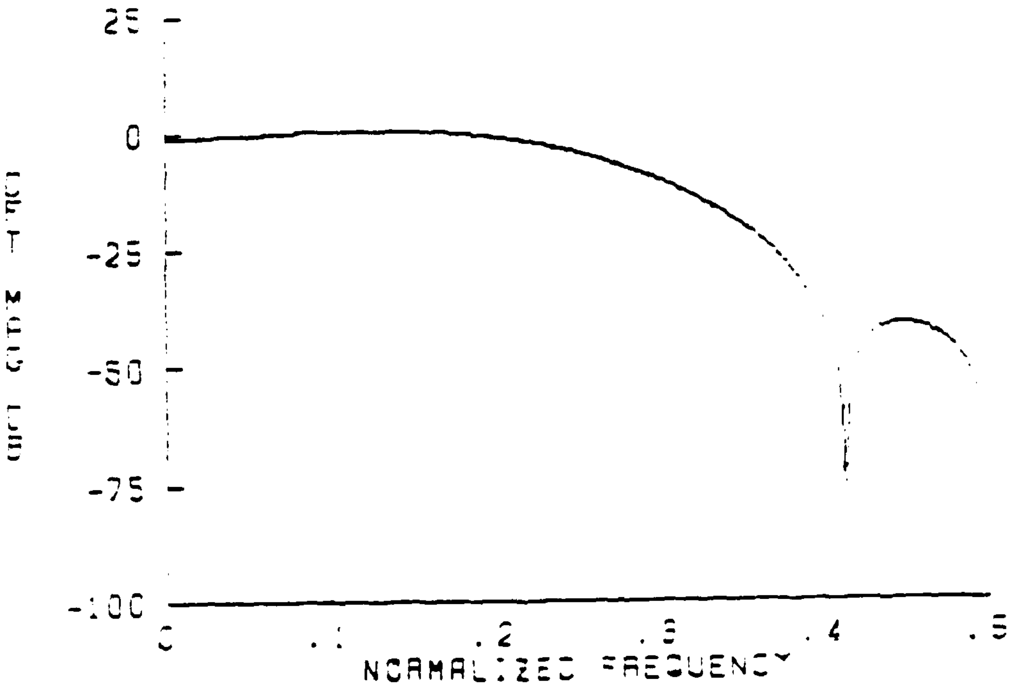


Figure 5.1b Frequency response of eight-tap unknown filter

orthogonal transform for each run we could measure several things: 1) the convergence of each transform versus an untransformed input, and 2) the best overall transform. The first was necessary to illustrate that in no case would the results of transforming the input be worse than no transform at all. The second has to do with the robustness of the transforms. It has been shown that for a given input, there is an optimal transform that will result in a complete decorrelation of the R_{xx} matrix. For any such input this is the Karhunen-Loève Transform, which has been shown to be unsuitable for our purposes because it is not a time-invariant transform. By requiring a time-invariant transform, one may find behavior that is not consistent for all inputs. For example, it has been shown that for a stationary sample of speech, the Discrete Cosine Transform will most effectively decorrelate the signal [8]. One possible explanation for this is that the eigenvectors of speech may line up at almost the same directions as the rows of the DCT, while the same is not true for the other transforms tested. This does not mean the DCT is necessarily the transform to use for signals with statistics unlike speech.

If the input is white noise, it was previously demonstrated that any of the transformed inputs will converge as fast as, but no faster than, the time domain case [7]. This comes about because the already random vector gains nothing from the orthogonalization process.

To compare the performance of the transforms and the time domain, the adaptive filters of Figures 5.2 and 5.3 were simulated, using the low-pass system model described above as the desired signal, and each of the four colored input

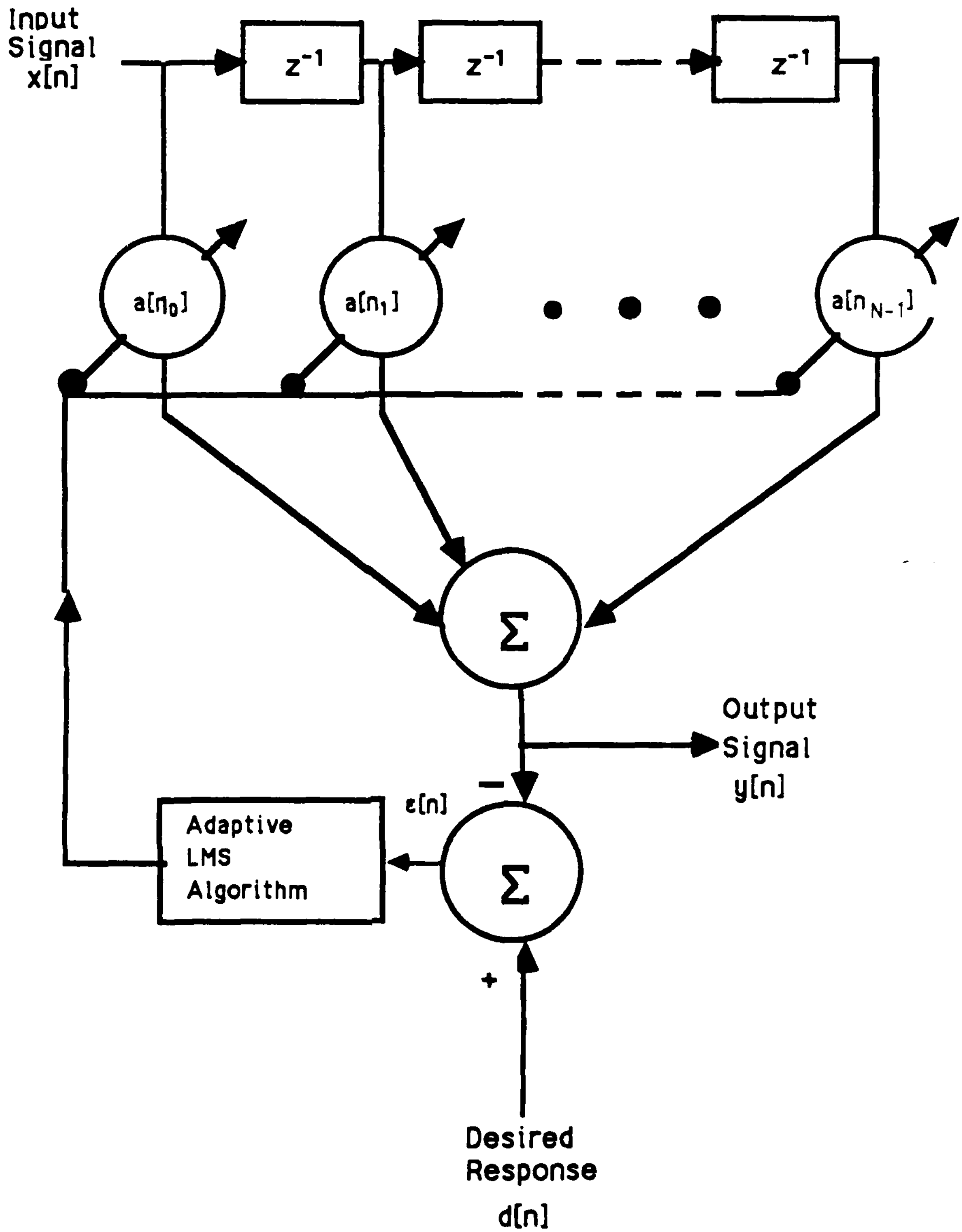


Figure 5.2

The tapped delay line adaptive filter

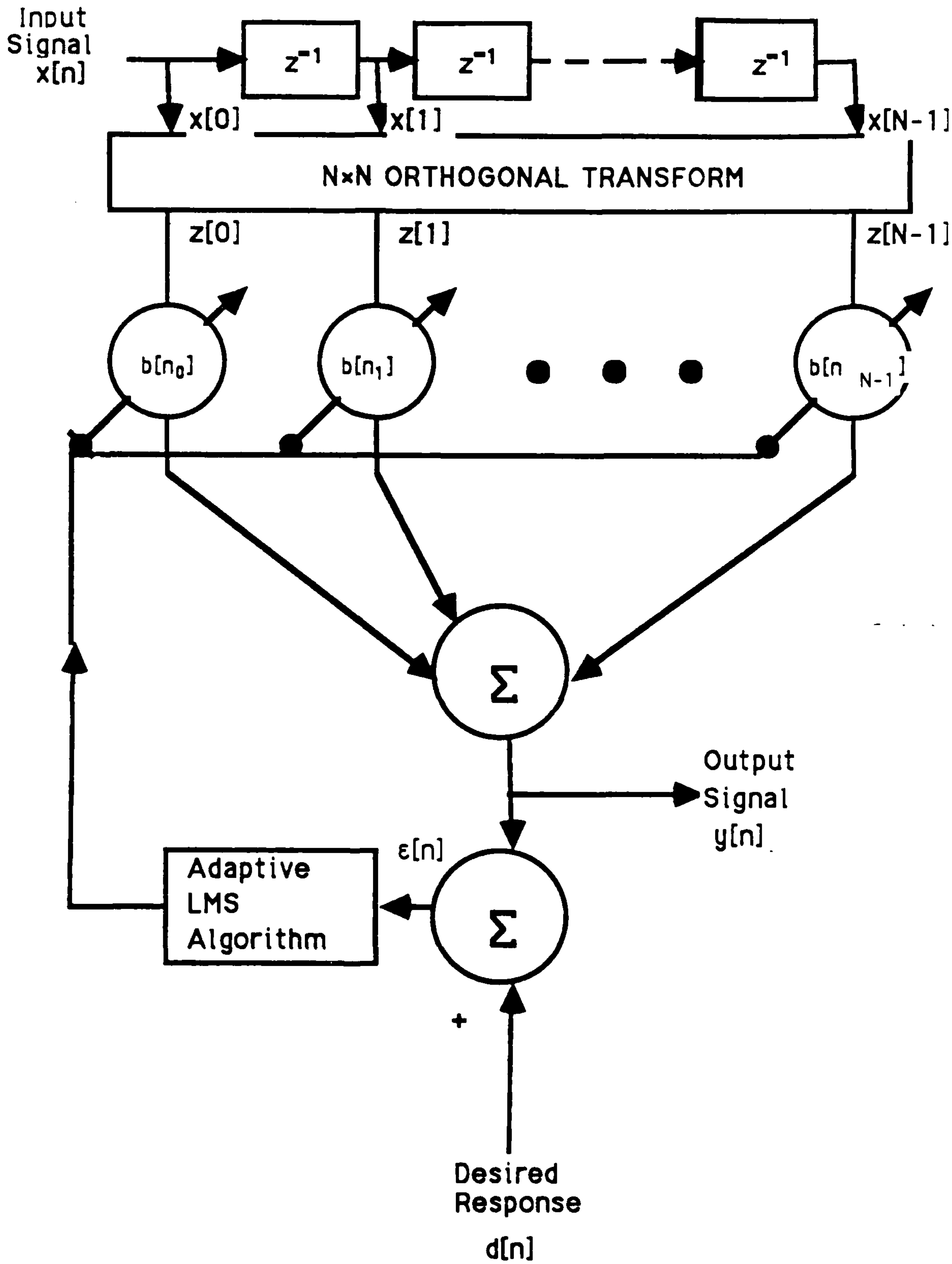


Figure 5.3

An FIR adaptive filter using an orthogonal transform to improve convergence

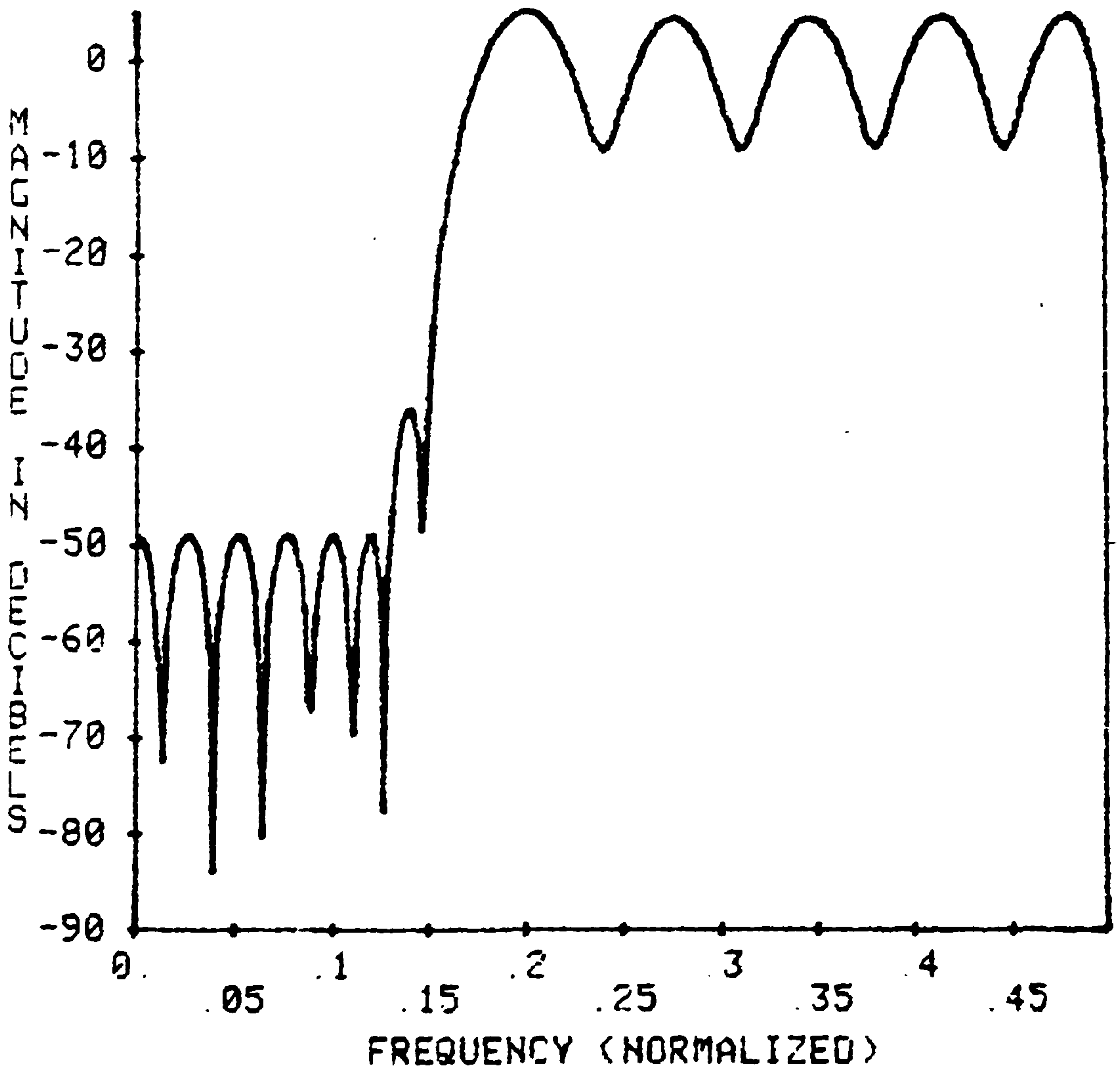


Figure 5.4. The frequency response of the highpass coloring filter

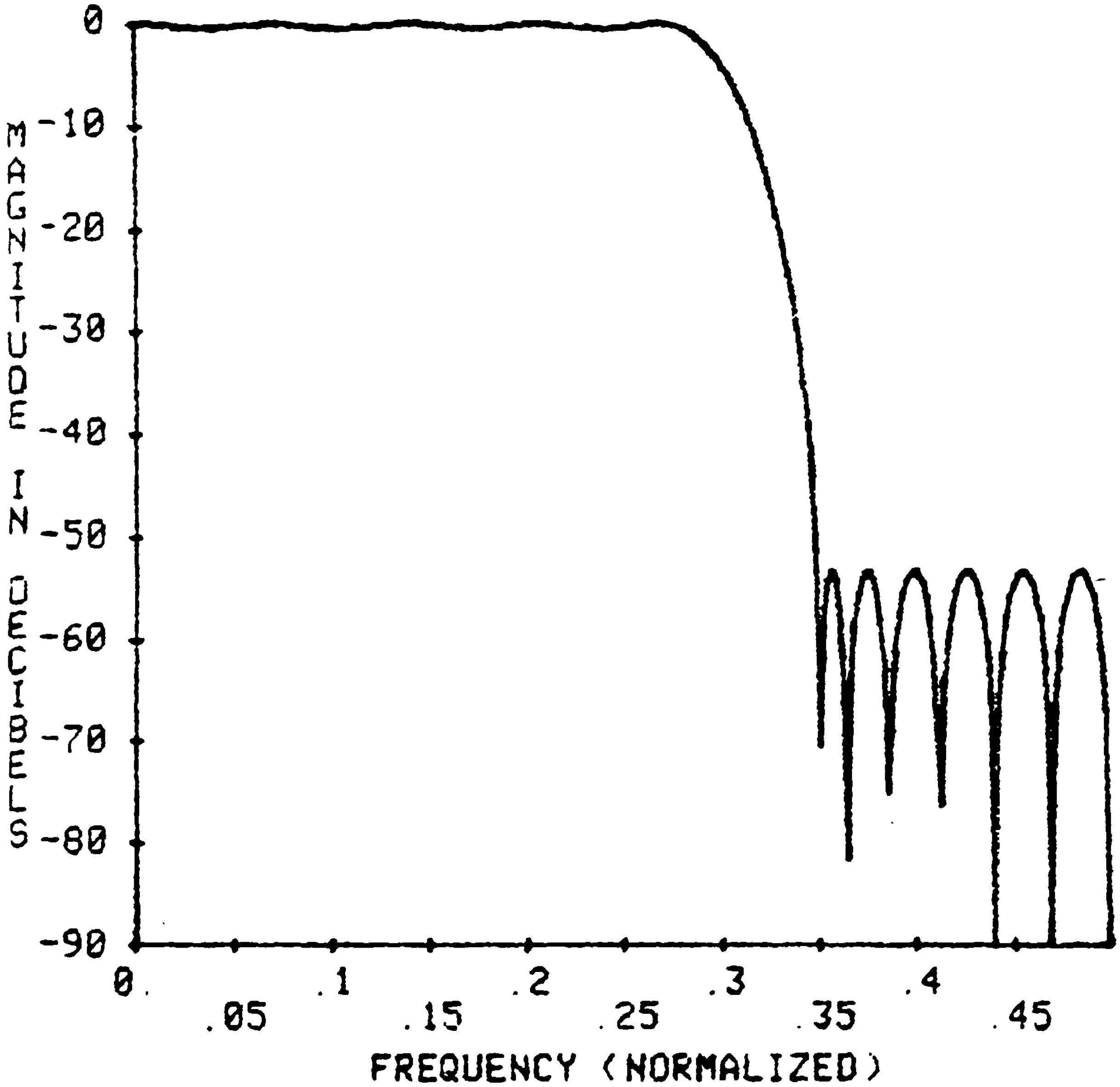


Figure 5.5 The frequency response of the low-pass coloring filter

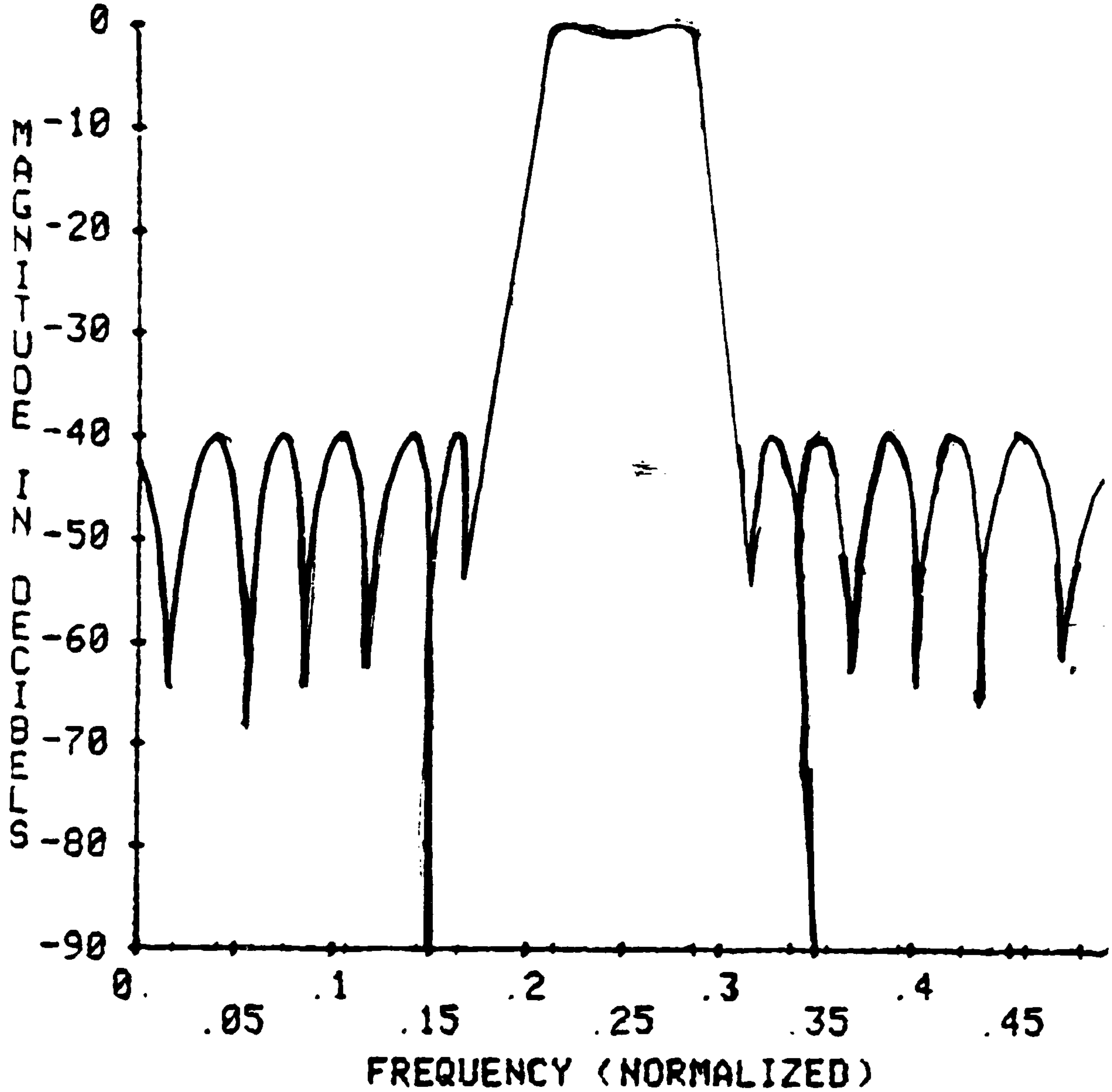


Figure 5.6 The frequency response of the band-pass coloring filter

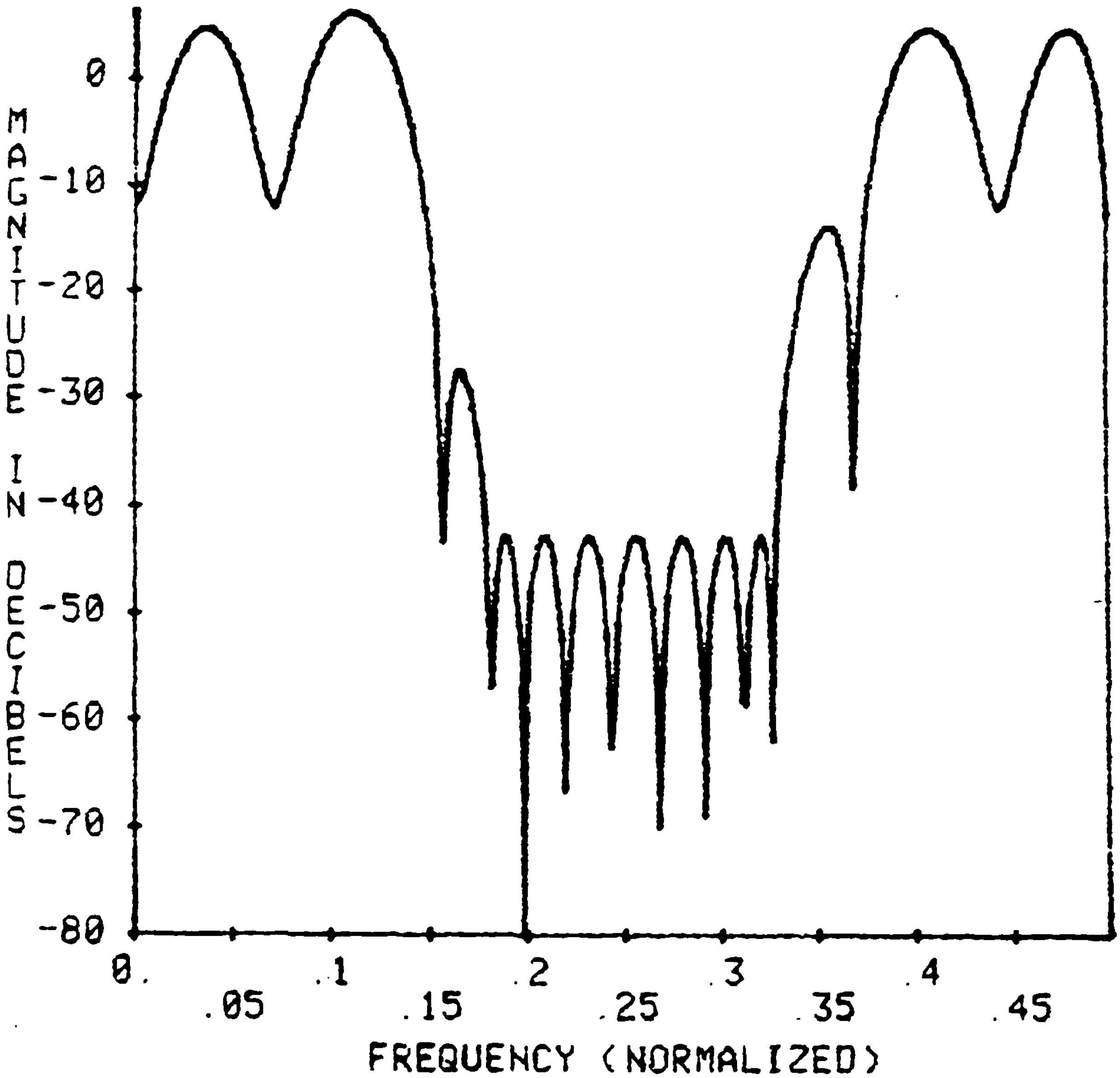


Figure 5.7 The frequency response of the band-reject coloring filter

spectra shown in Figures 5.4-5.7 to shape the input spectrum. The adaptive system is configured in the system identification mode; thus, the final output of the system will be a set of adapted coefficients, $A(\infty)$, representing the estimate of model spectra, with the MMSE resulting from the adaption representing the final misadjustment of those coefficients. In most cases the filter was allowed to adapt for a sufficient number of iterations to reach convergence. Convergence here is defined as that point where the MSE has reached -300dB, the noise floor of the CYBER. In some instances, most notably several time-domain runs, the process has not yet converged at the end of 20,000 iterations. These processes were terminated before they became prohibitively expensive. In all instances, however, enough adaptation has occurred to establish a trend that can be carried to convergence if necessary.

The result that was used to judge the effectiveness of the transforms was the adaptive learning curve, a measurement of the MSE versus the number of iterations. Another factor that may prove useful in future studies is the actual set of filter coefficients generated by the transforms, which will show how well the adaptation worked in areas of different power densities.

5.3 Comparison with the Time Domain

The first colored input noise case is zero-mean white noise from RANF passed through a 32-tap linear FIR high-pass filter, whose frequency response is shown in Figure 5.4., then fed to the time and transformed adaptive systems.

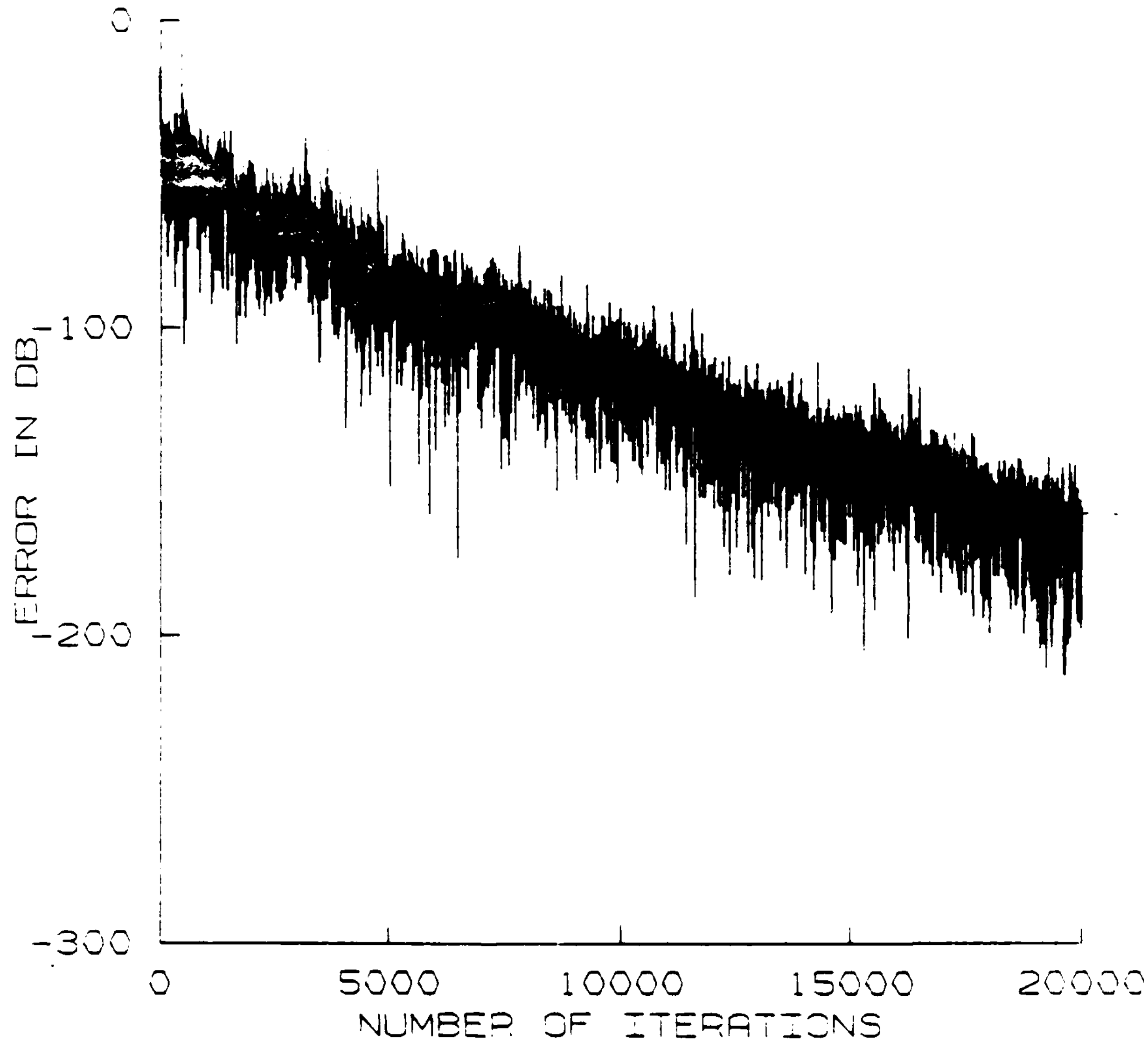


Figure 5.8 The learning curve for the time-domain filter trained with high-pass noise

The system was first adapted using the time-domain LMS algorithm. The learning curve for this process is shown in Figure 5.8. This curve exhibits quite a bit of variation because it was not smoothed, but the downward trend is obvious. One can immediately see that this system has not converged to the noise floor even after 20,000 iterations.

To compare each of the transforms to the time domain, the transform adaptive filter was trained with the high-passed noise. The same algorithm was repeated five times, once with complex arithmetic for the DFT, and four times using real arithmetic for the DCT, DHT, WHT, and PO2. The only difference between runs was the substitution of each transform in the signal path. The learning curves for each transform are shown individually, superimposed over the time domain in Figures 5.9-5.13. Clearly, these curves for the transformed inputs descend more steeply, which translates to fewer iterations before convergence. Thus, all the transformed adaptive filters can be said to outperform the time domain filter for this type of colored noise input.

The process of comparing the time domain to the transformed domains was repeated using three more coloring filters, low-pass, band-pass, and band-reject, each a 32-tap linear FIR filter created with DFDZR1. The frequency responses of each are shown in Figures 5.5-5.7. A time domain adaptation learning curve was generated for each colored input, shown in Figures 5.14-5.16. The low-pass and band-pass cases are very similar to the high-pass; neither one has converged after 20,000 iterations. The band reject case is strikingly different. Not only has the time-domain filter converged, it has converged in what appears to be about 7500

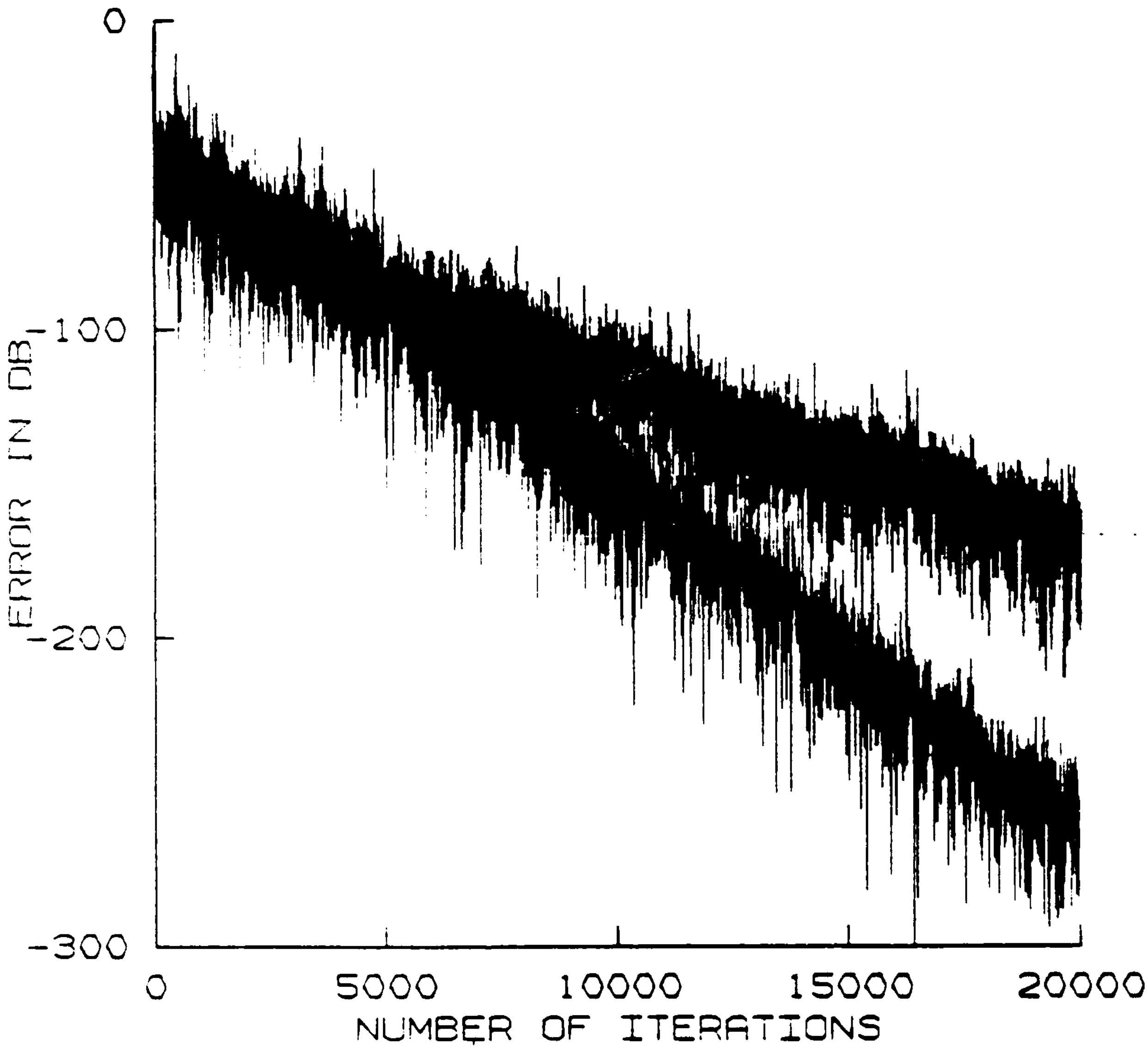


Figure 5.9 The DFT versus time for high-pass noise

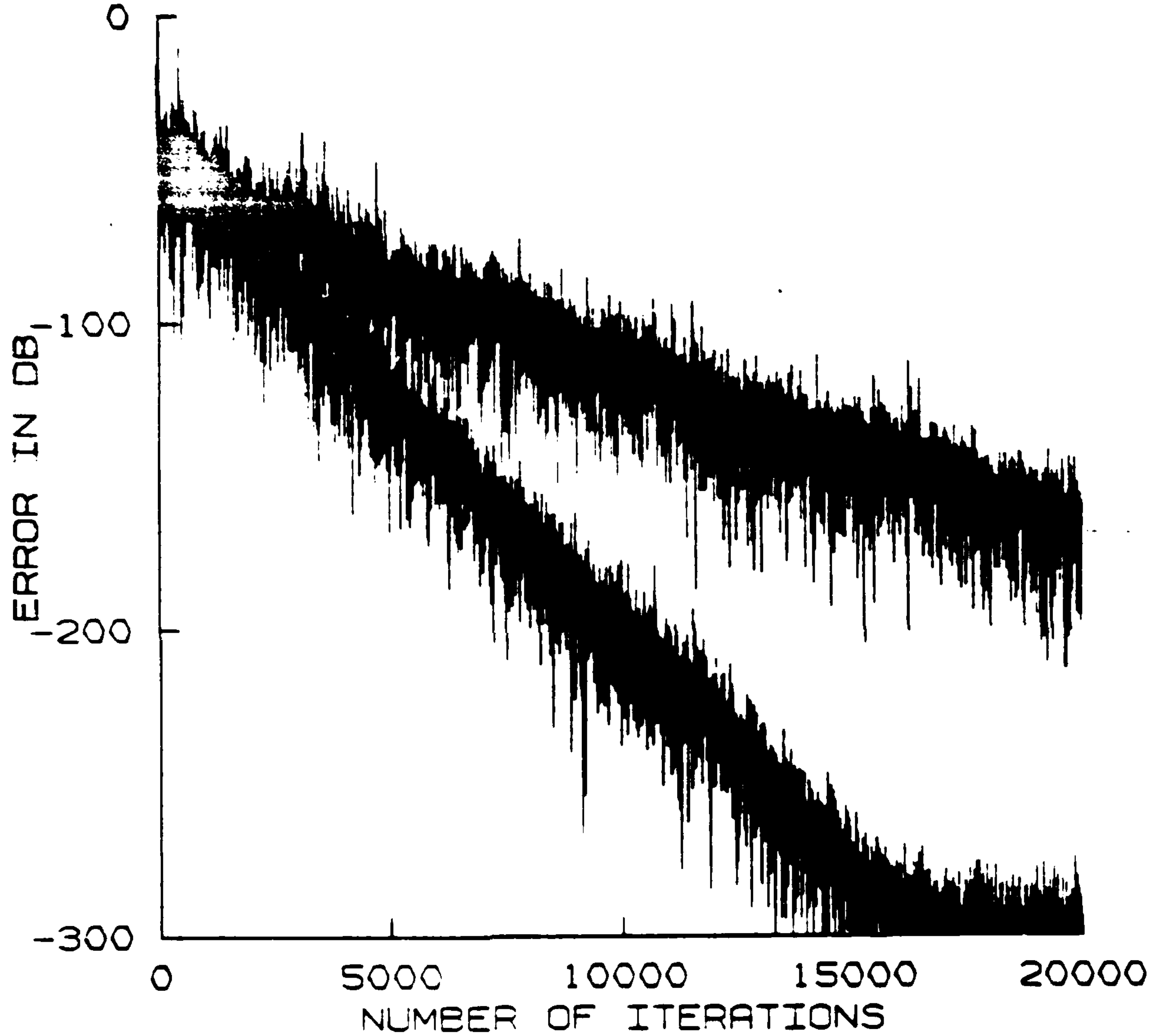


Figure 5.10 The DCT versus time for high-pass noise

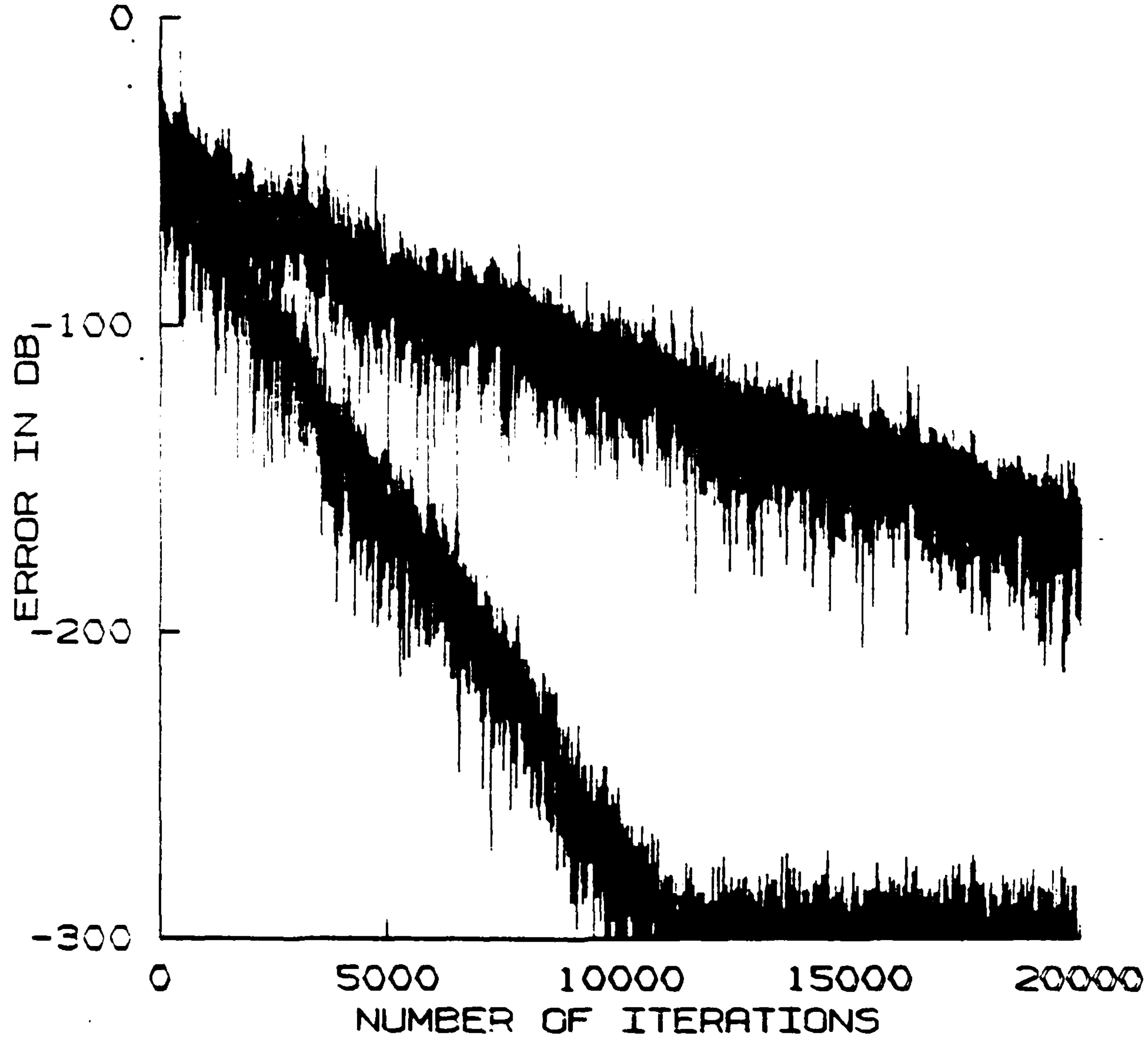


Figure 5.11 The DHT versus time for high-pass noise

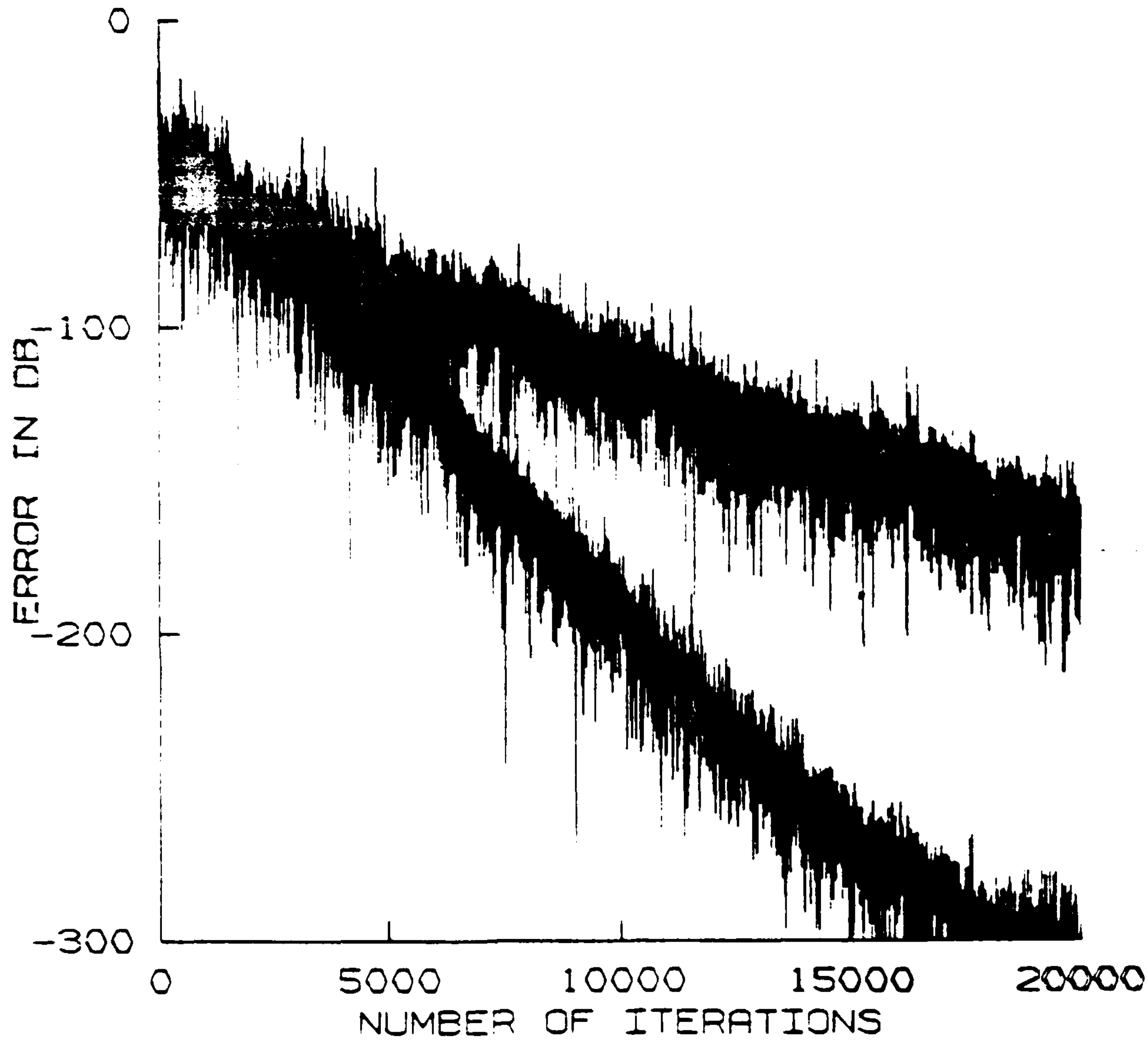


Figure 5.12 The WHT versus time for high-pass noise

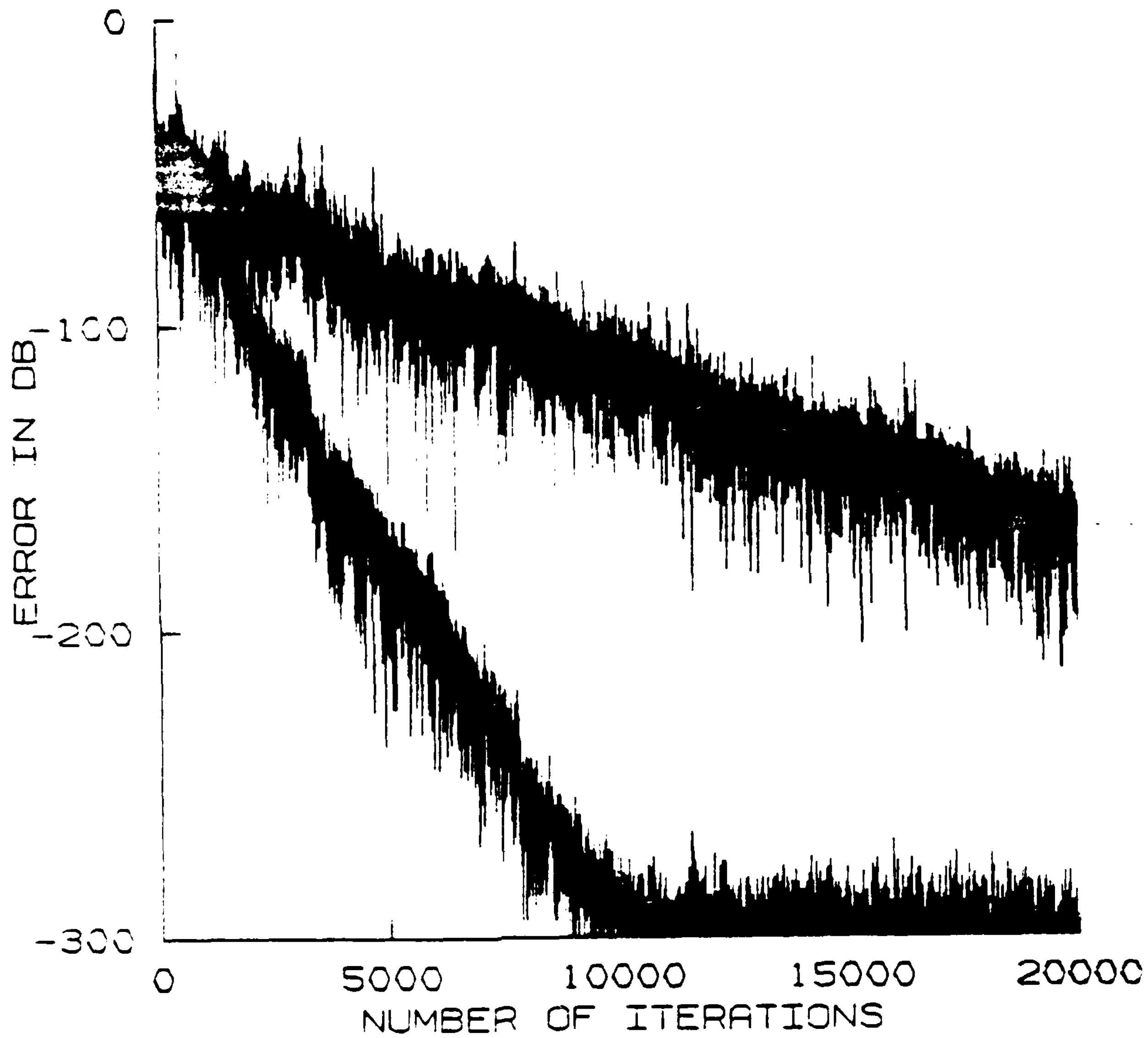


Figure 5.13 The PO2 versus time for high-pass noise

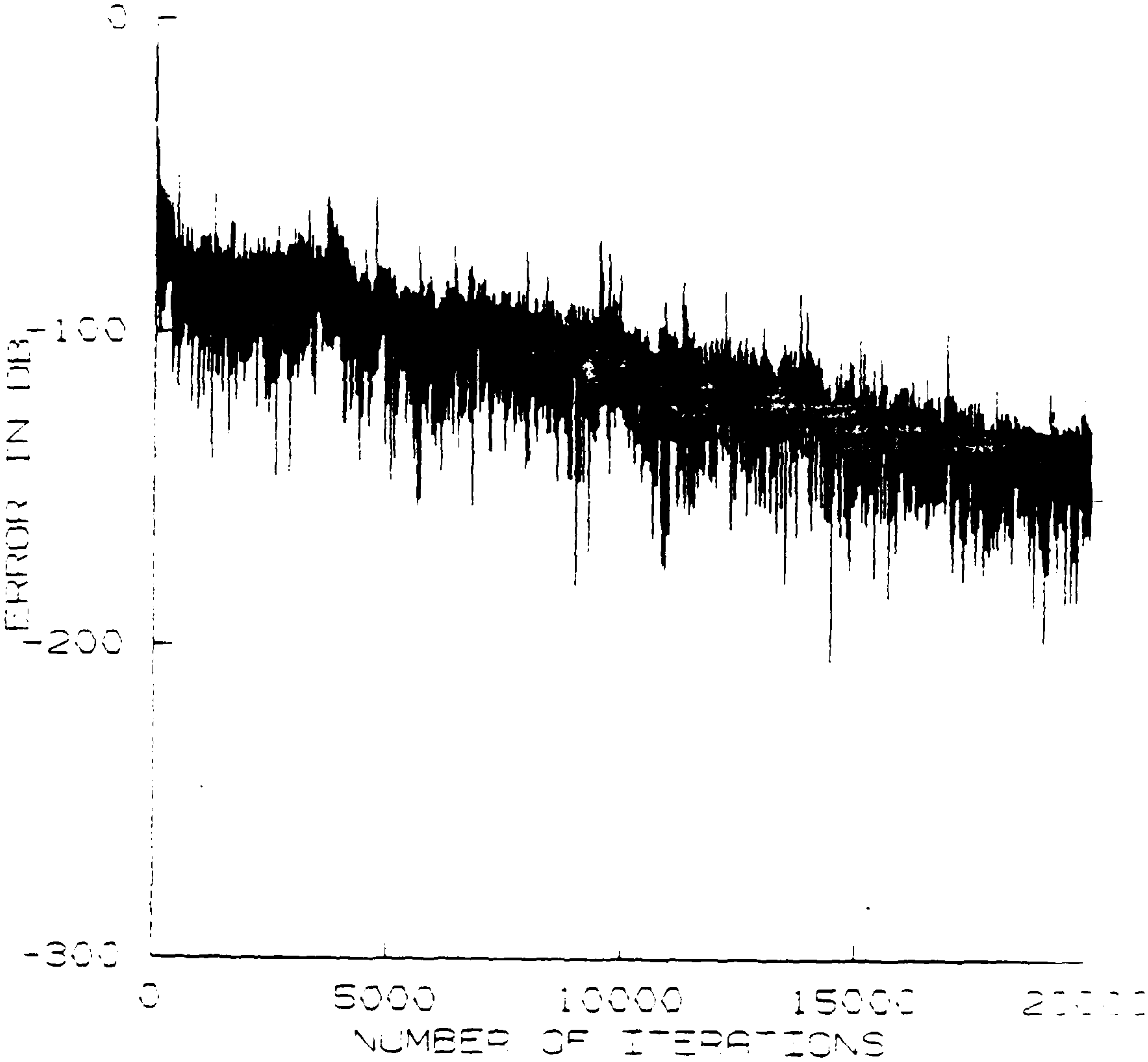


Figure 5.14 The learning curve for the time-domain filter trained with low-pass noise

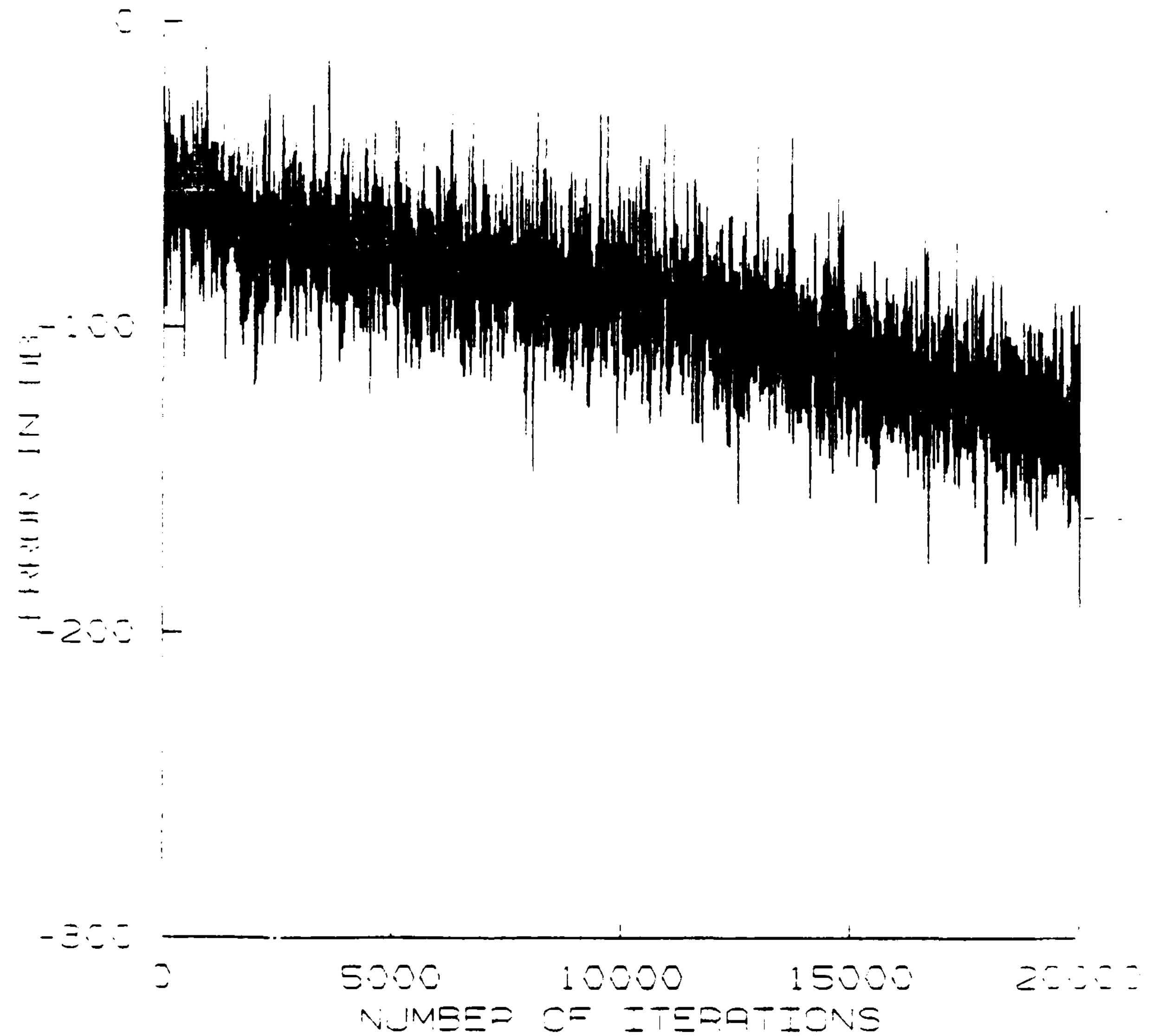


Figure 5.15 The learning curve for the time-domain filter trained with band-pass noise

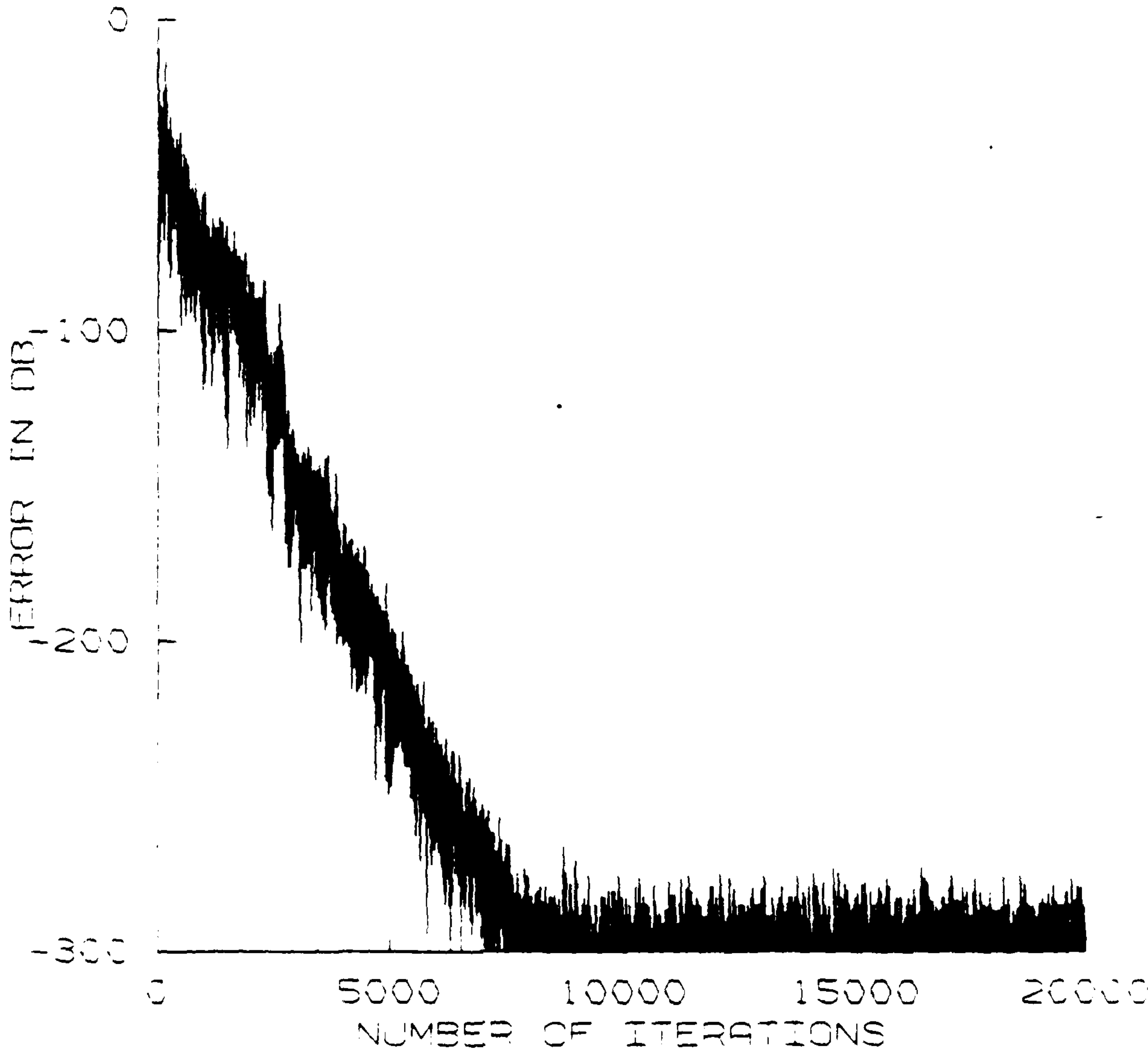


Figure 5.16 The learning curve for the time-domain filter trained with band-reject noise

iterations. This phenomenon will be commented on later.

Each transform was again compared to the time domain, with the same result as the first case. In each of these three cases, the transformed inputs required fewer iterations to reach convergence. Thus, all the transformed adaptive filters can be said to outperform the time domain filter for these types of colored noise input. The series of figures showing the resulting learning curves of each transform over the corresponding time domain learning curve begins with Figures 5.17-5.21 for the low-pass case, and includes Figures 5.22-5.26 for the band-pass case, and Figures 5.27-5.31 for the band-reject case.

Since the transformed filters perform better than the time domain for each of these four well known filter types, which in these experiments represent the shapes of input spectra, and have already been shown to perform as well with white noise, which could be interpreted as the fifth common filter type, the all-pass network, we can conclude that for any practical system the transform domain adaptive filter will be superior. As a point of theory, it is possible to construct a transform that, given a certain input, will actually display a learning curve inferior to the time domain. This theoretical transform has been shown here experimentally to be not one of any of the transforms we have examined. As it would seem that the experiment accounts for most, if not all, of the likely combination of transforms and inputs, this theoretical point is of little practical significance.

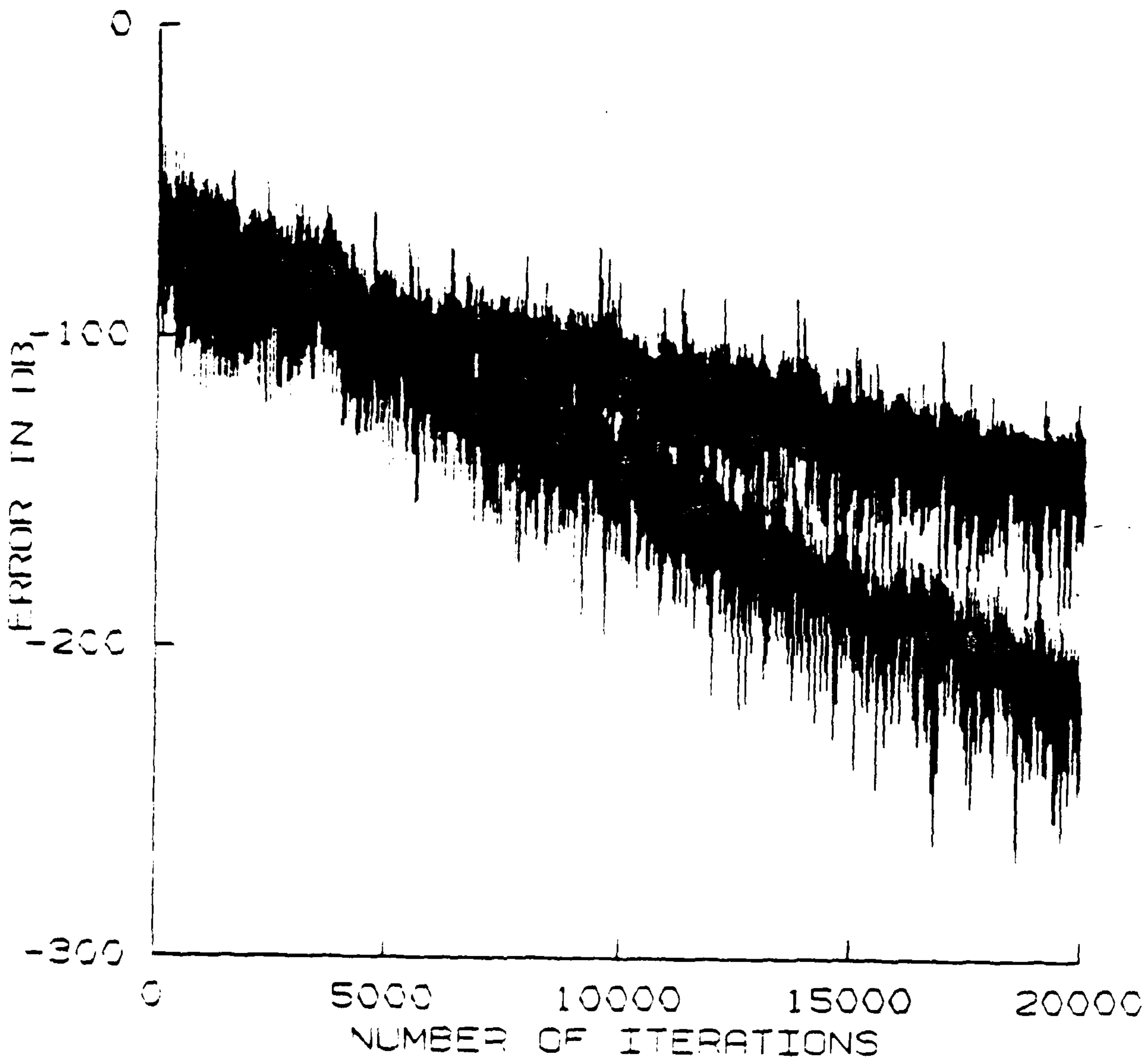


Figure 5.17 The DFT versus time for low-pass noise

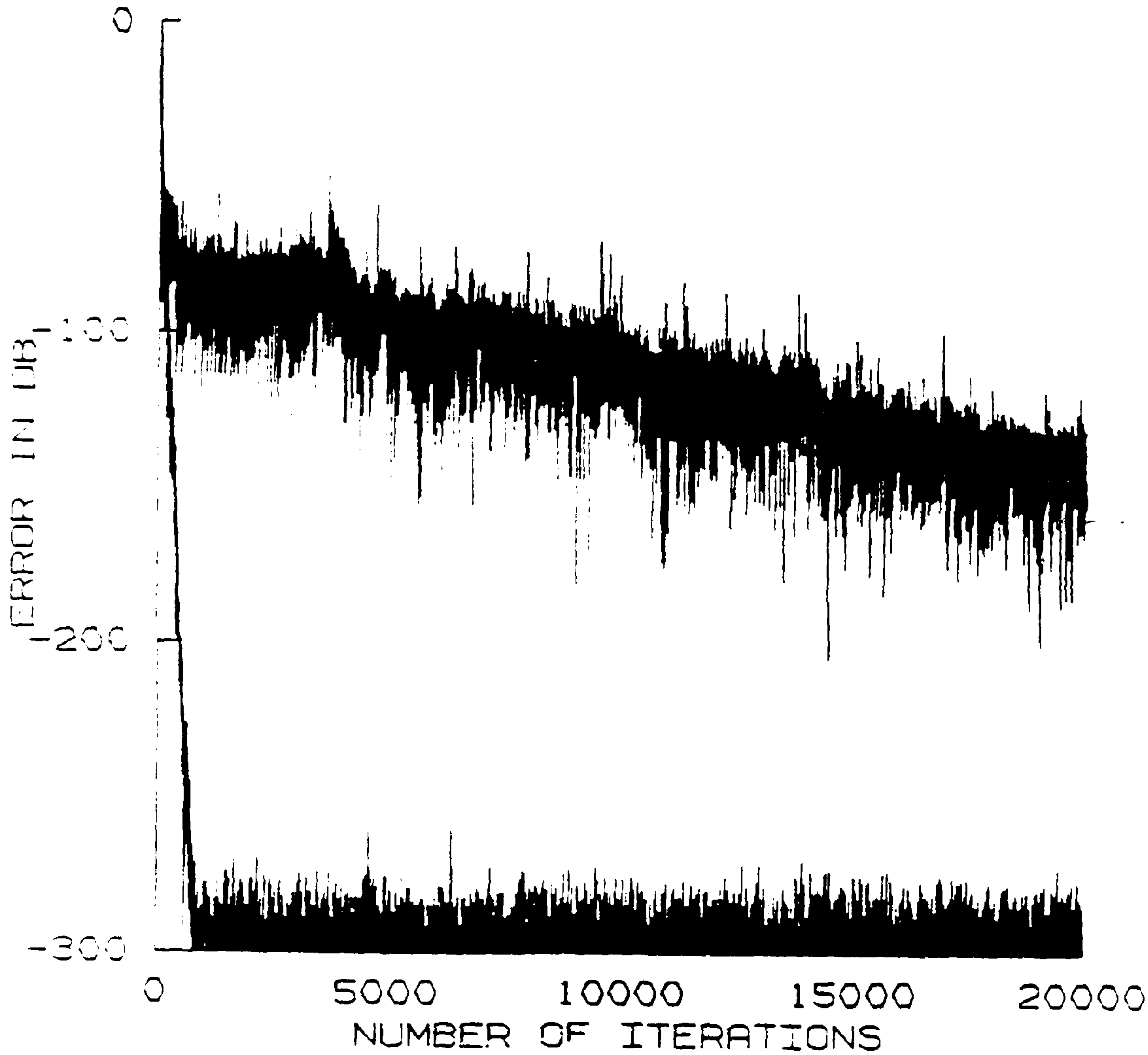


Figure 5.18 The DCT versus time for low-pass noise

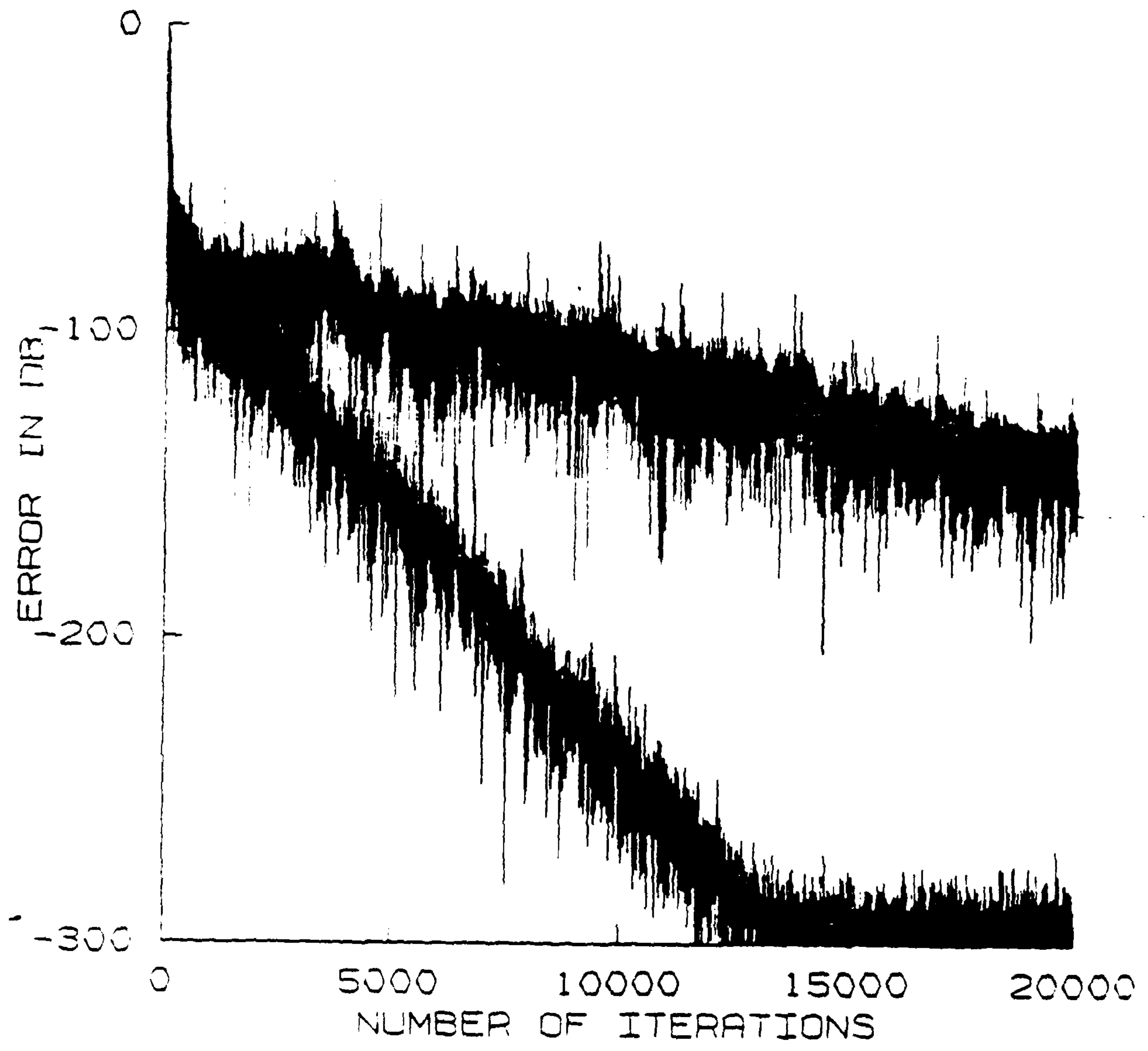


Figure 5.19 The DHT versus time for low-pass noise

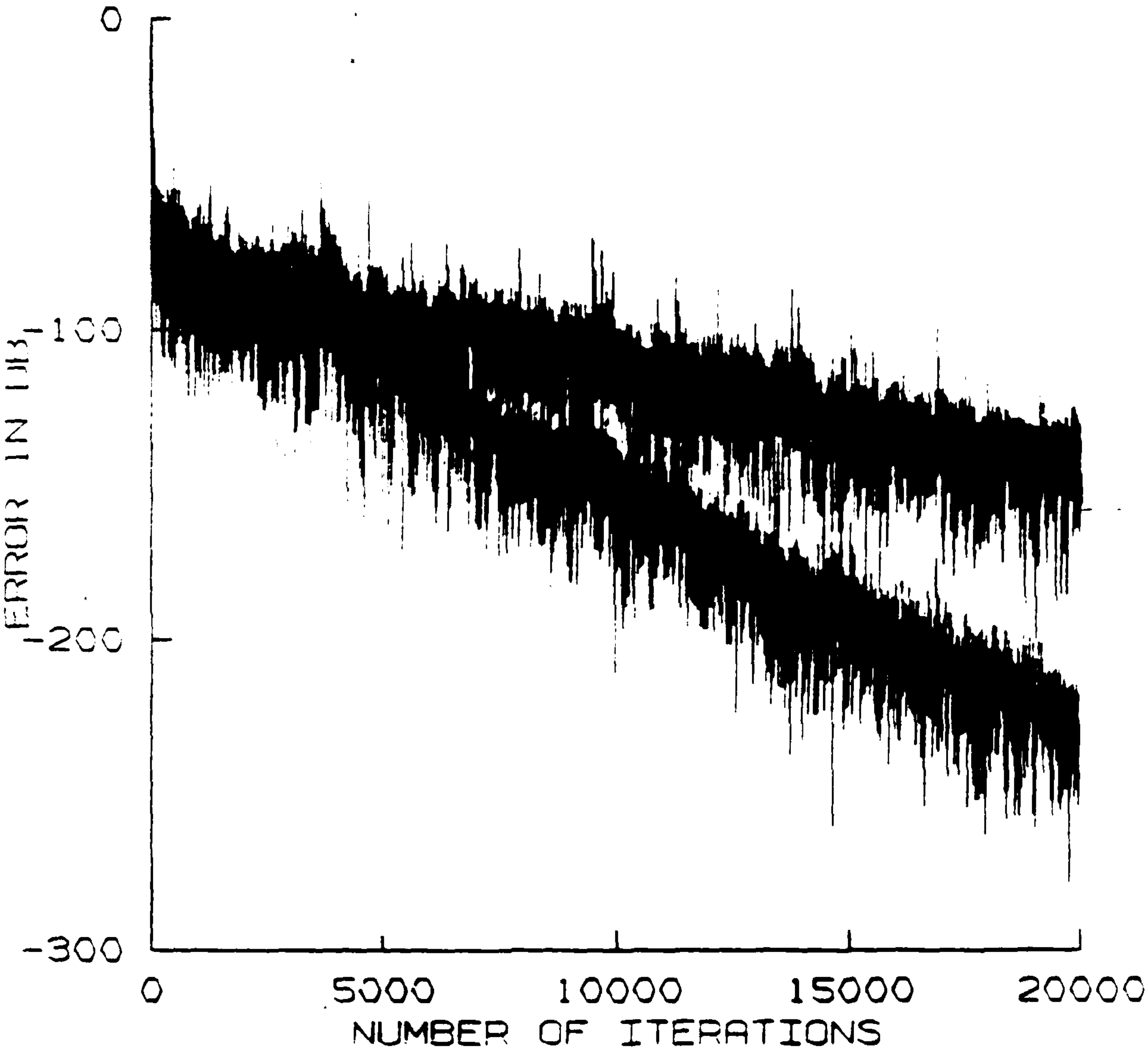


Figure 5.20 The WHT versus time for low-pass noise

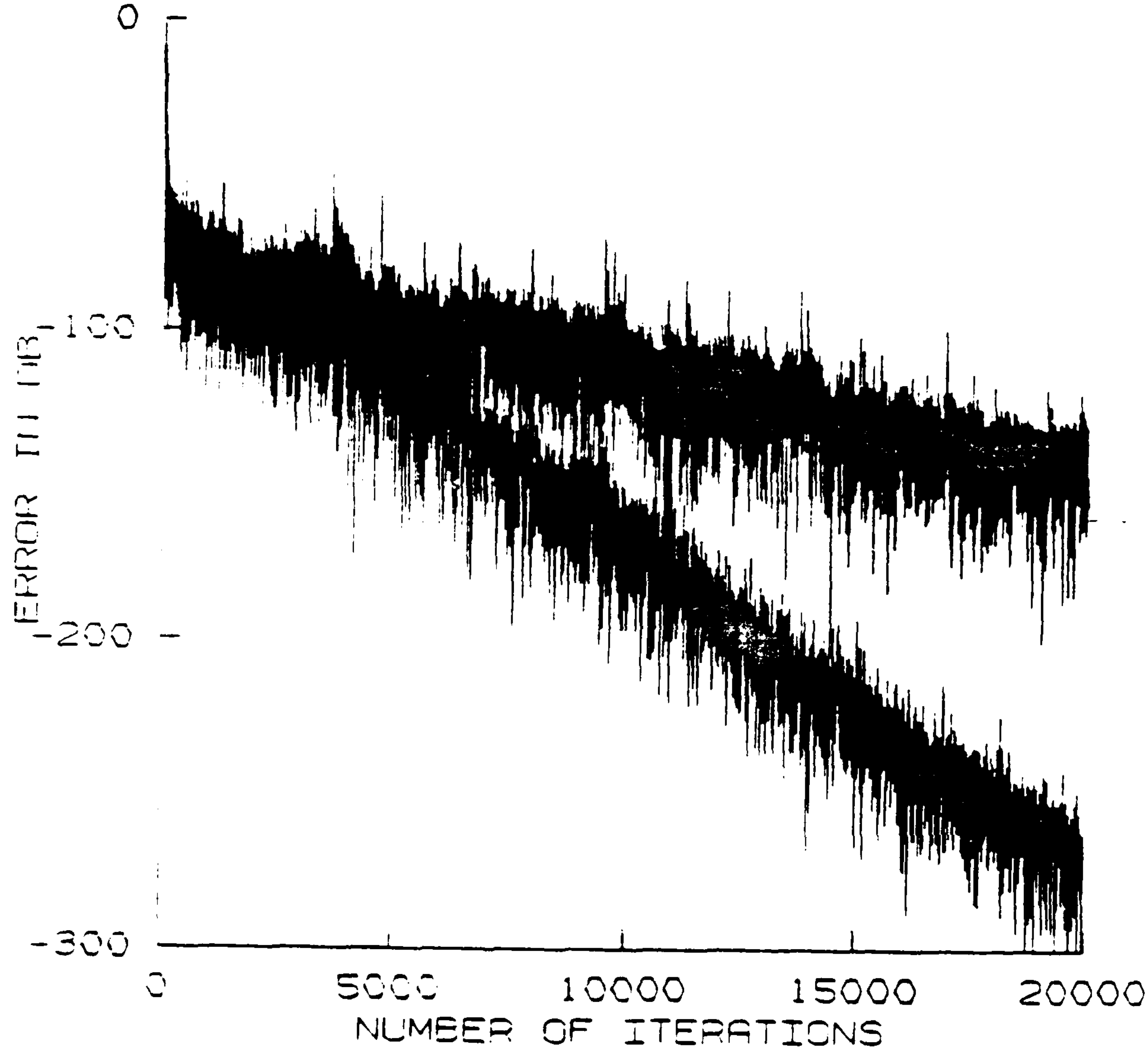


Figure 5.21 The PO2 versus time for low-pass noise

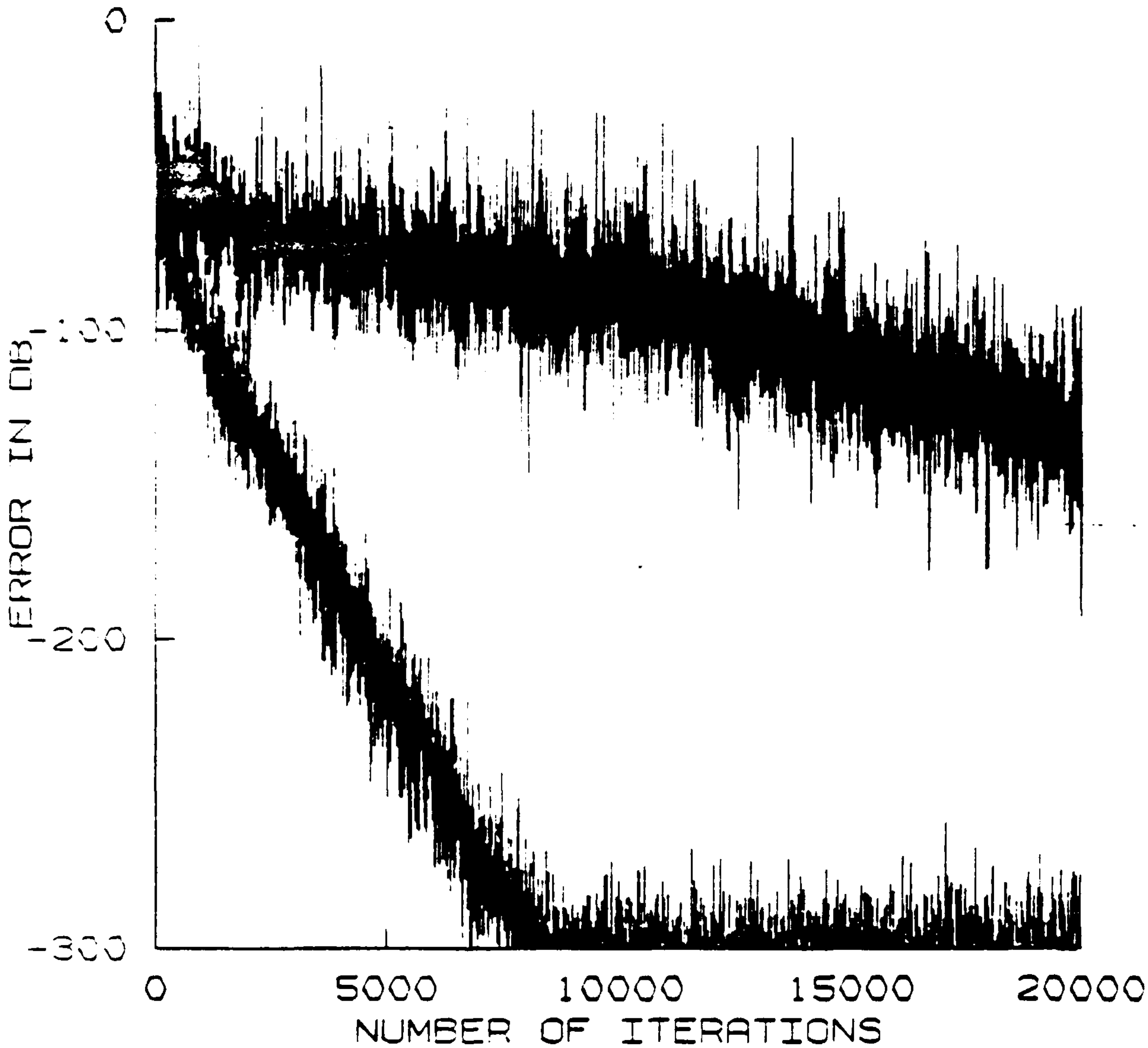


Figure 5.22 The DFT versus time for band-pass noise

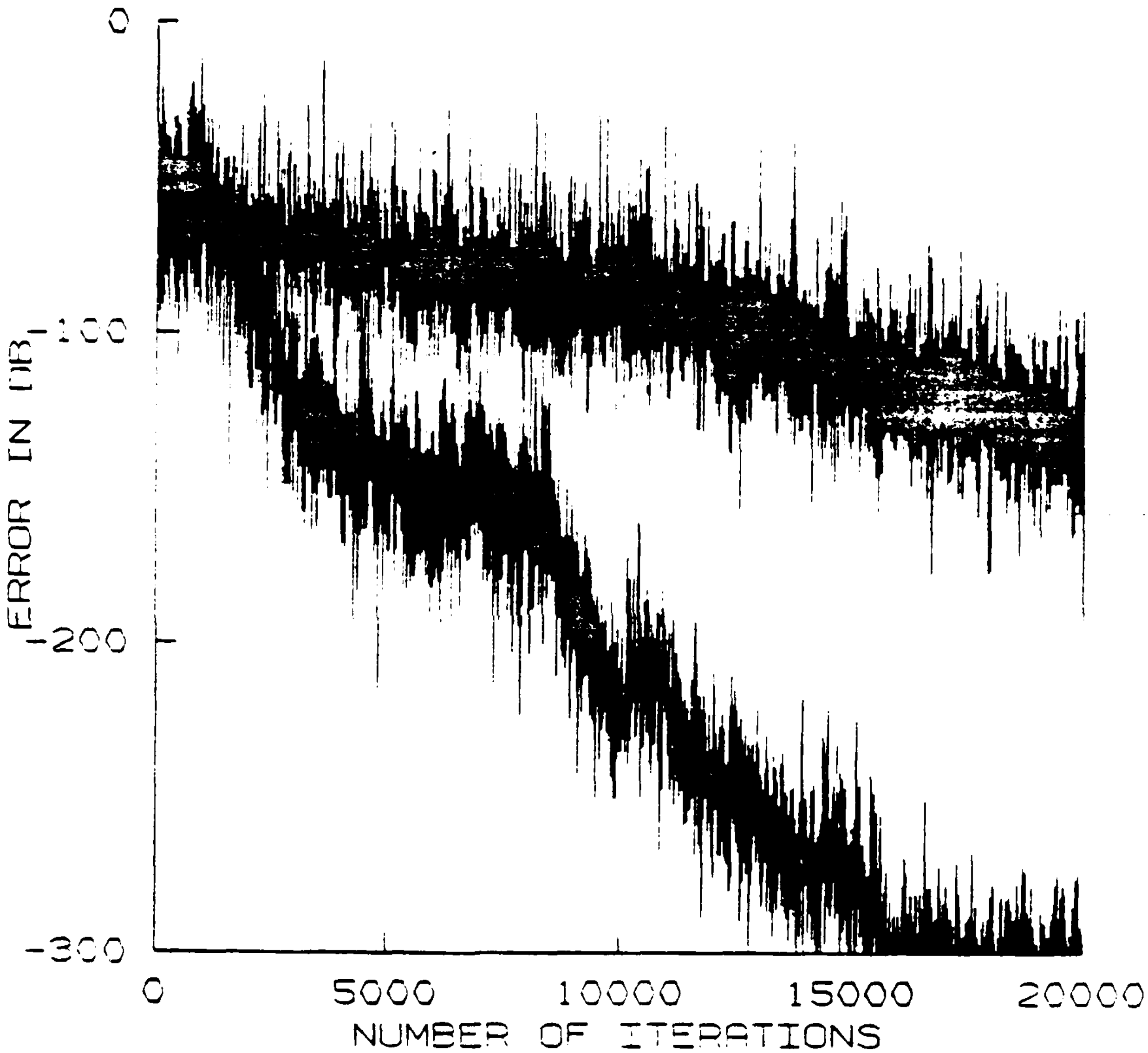


Figure 5.23 The DCT versus time for band-pass noise

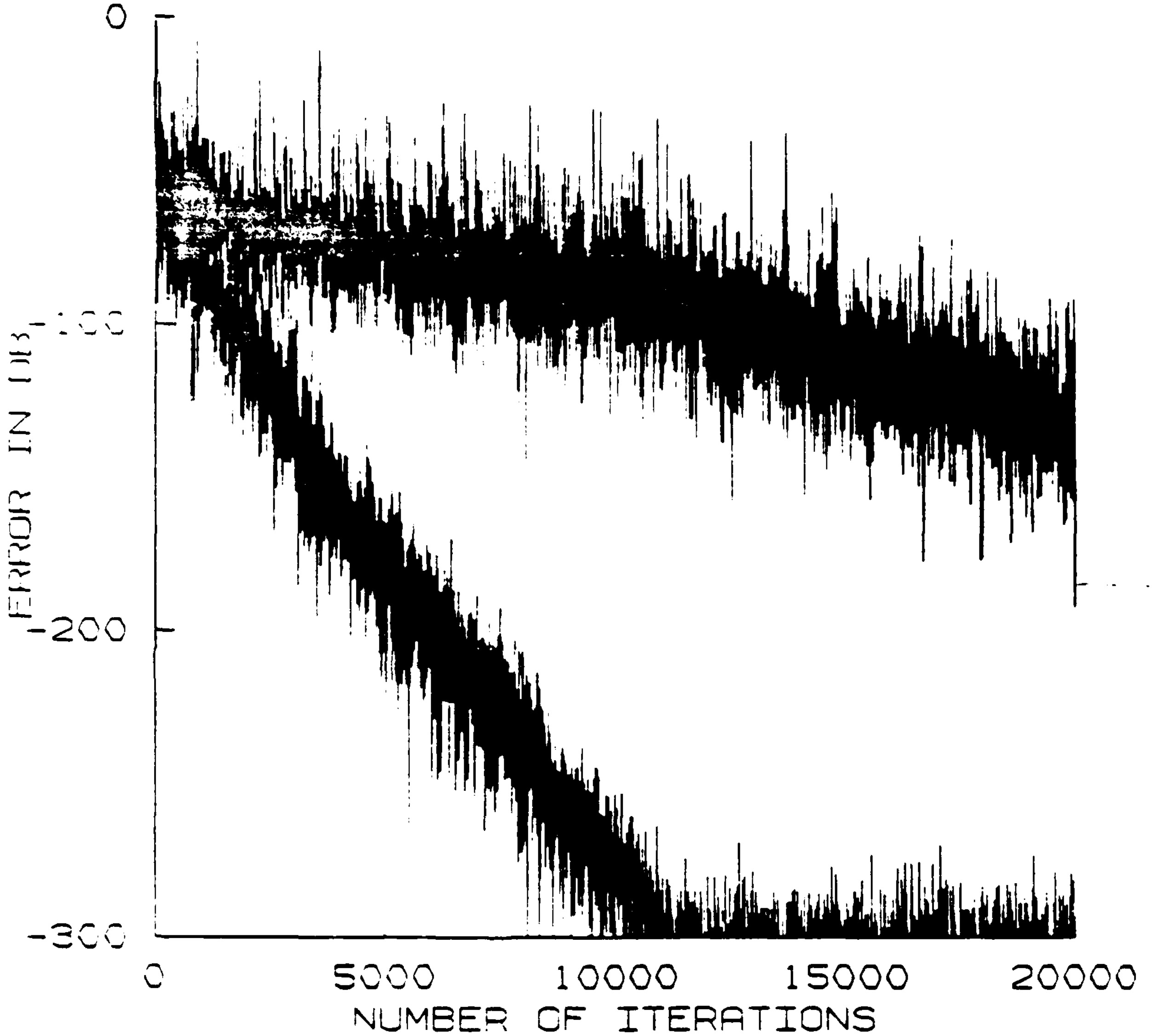


Figure 5.24 The DHT versus time for band-pass noise

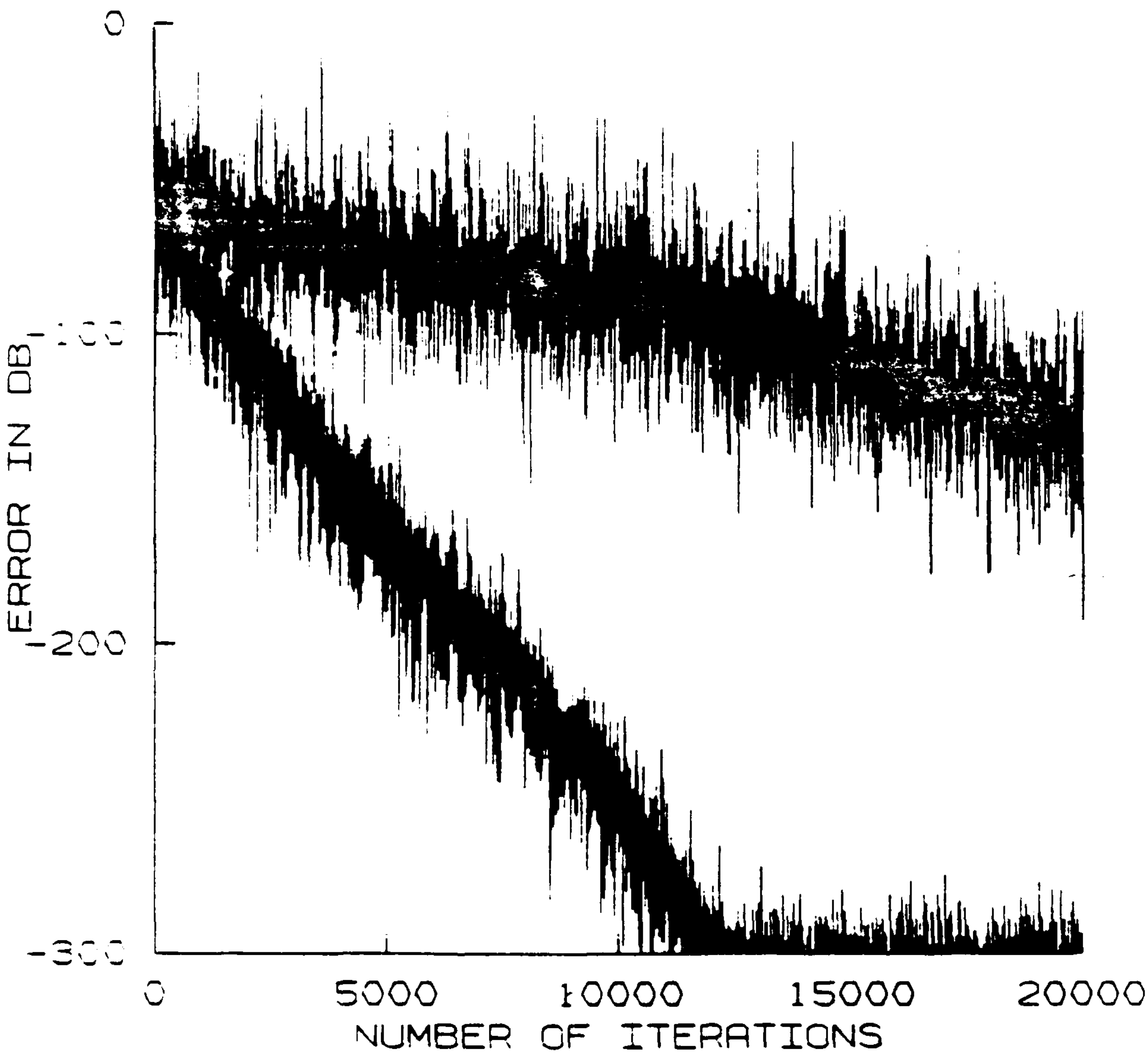


Figure 5.25 The WHT versus time for band-pass noise

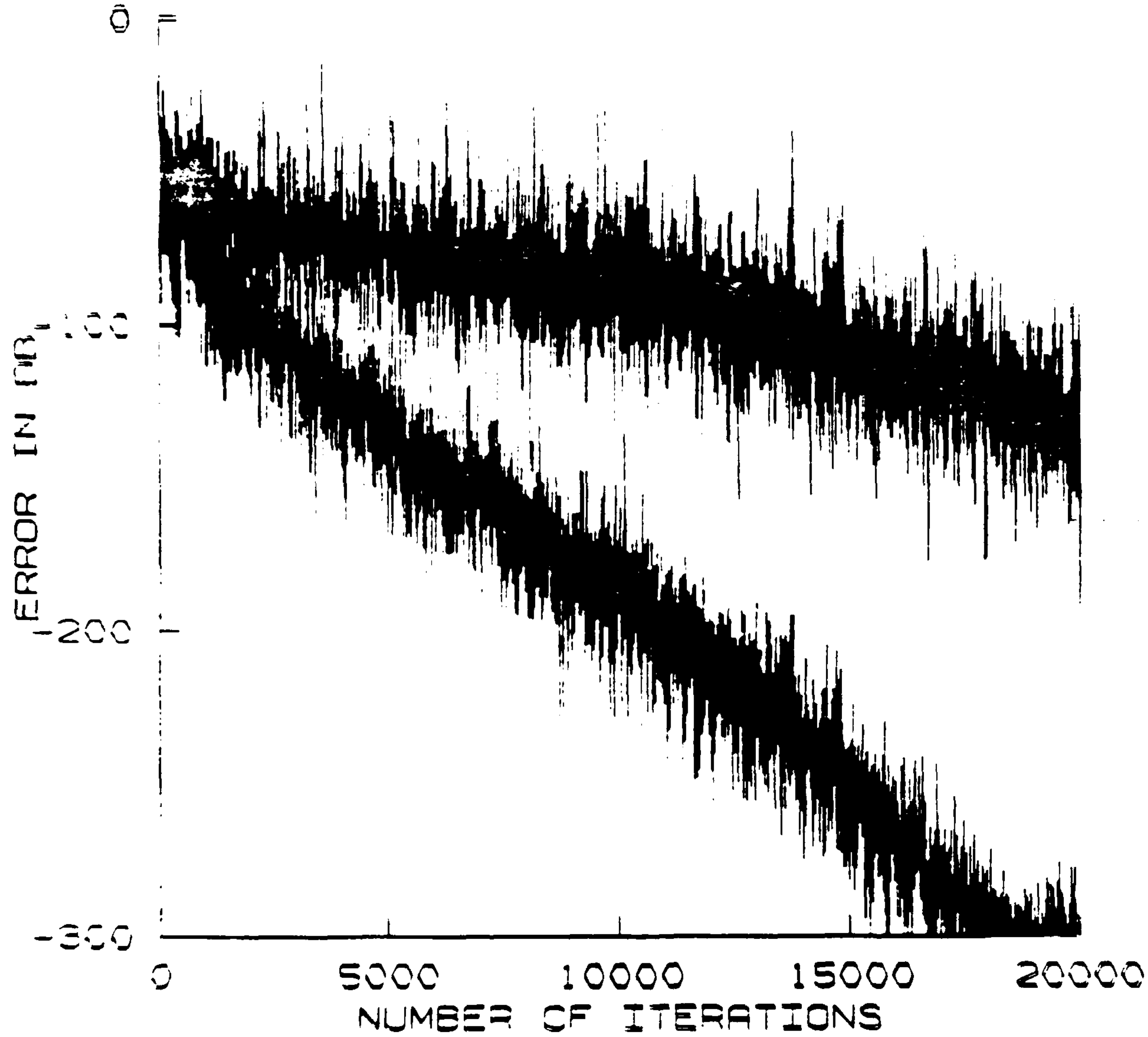


Figure 5.26 The PO2 versus time for band-pass noise

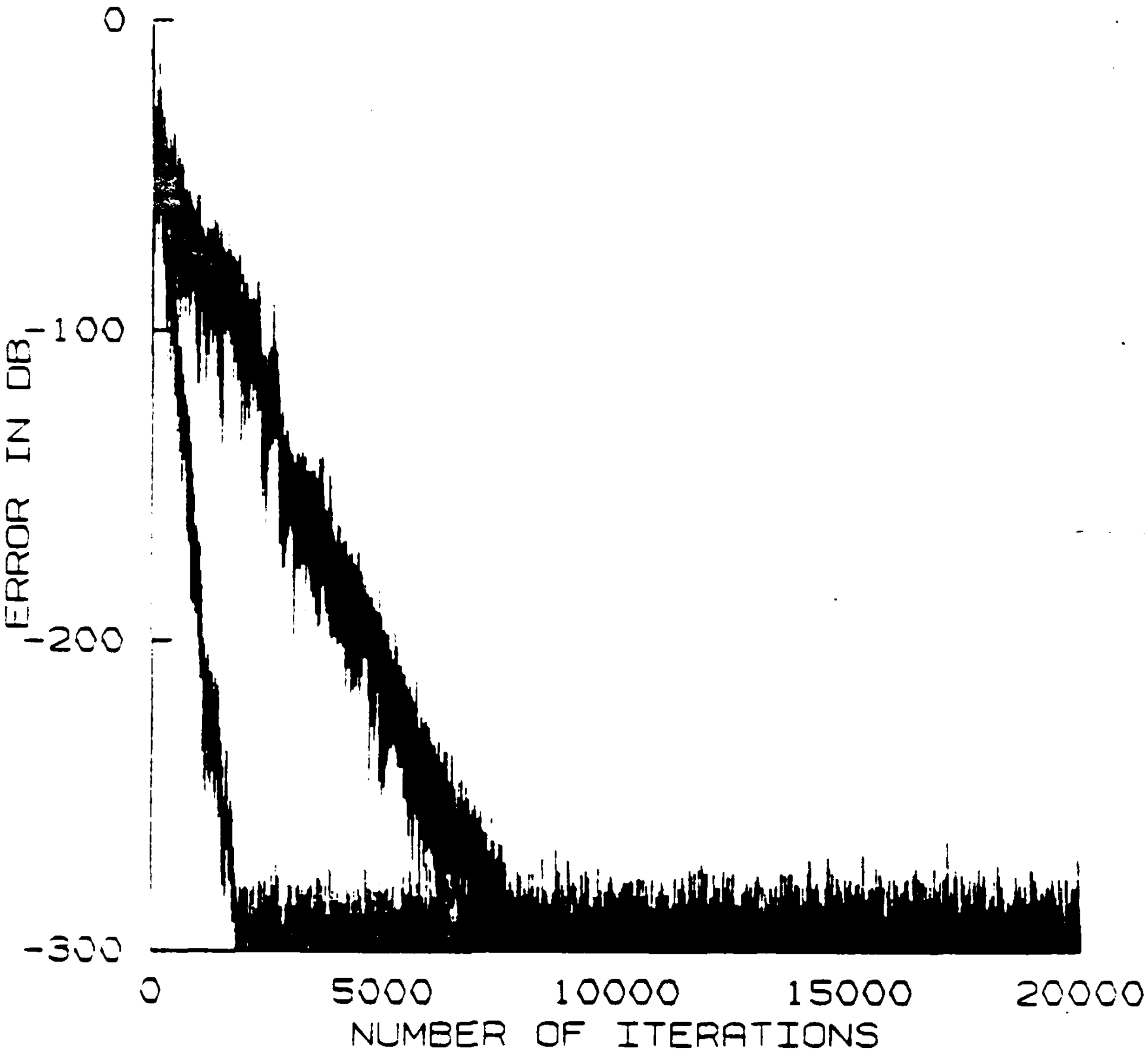


Figure 5.27 The DFT versus time for band-reject noise

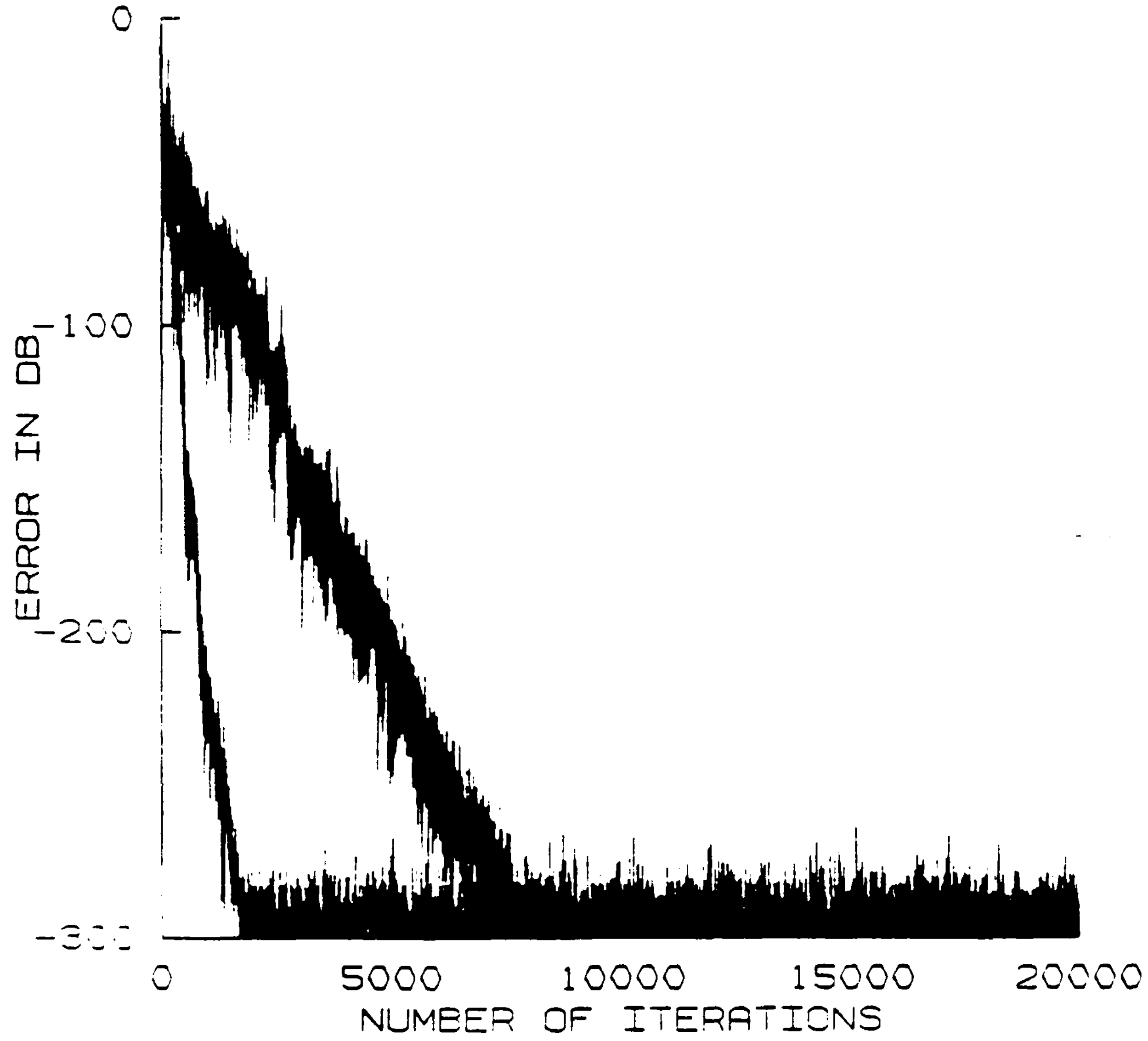


Figure 5.28 The DCT versus time for band-reject noise

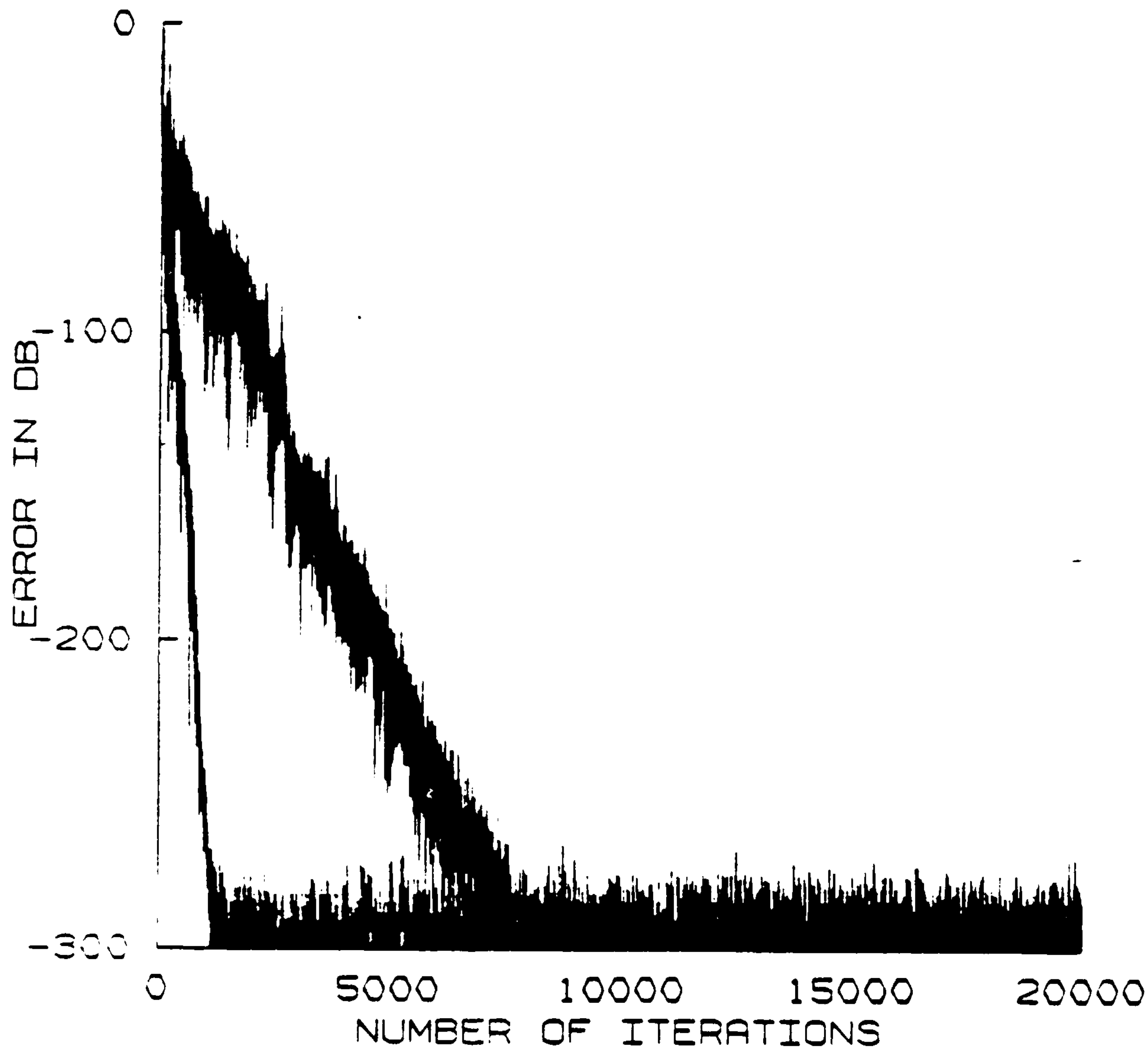


Figure 5.29 The DHT versus time for band-reject noise

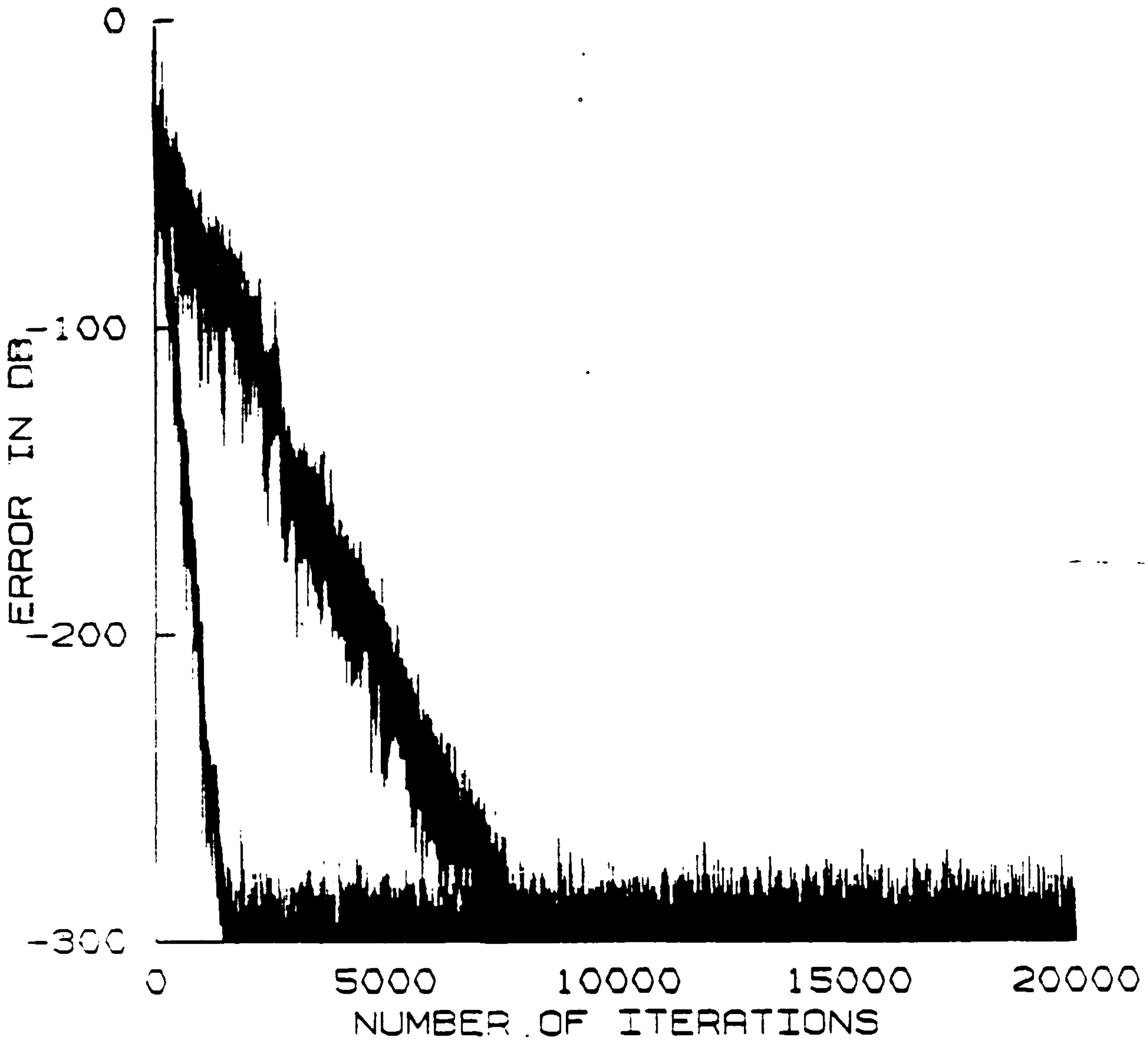


Figure 5.30 The WHT versus time for band-reject noise

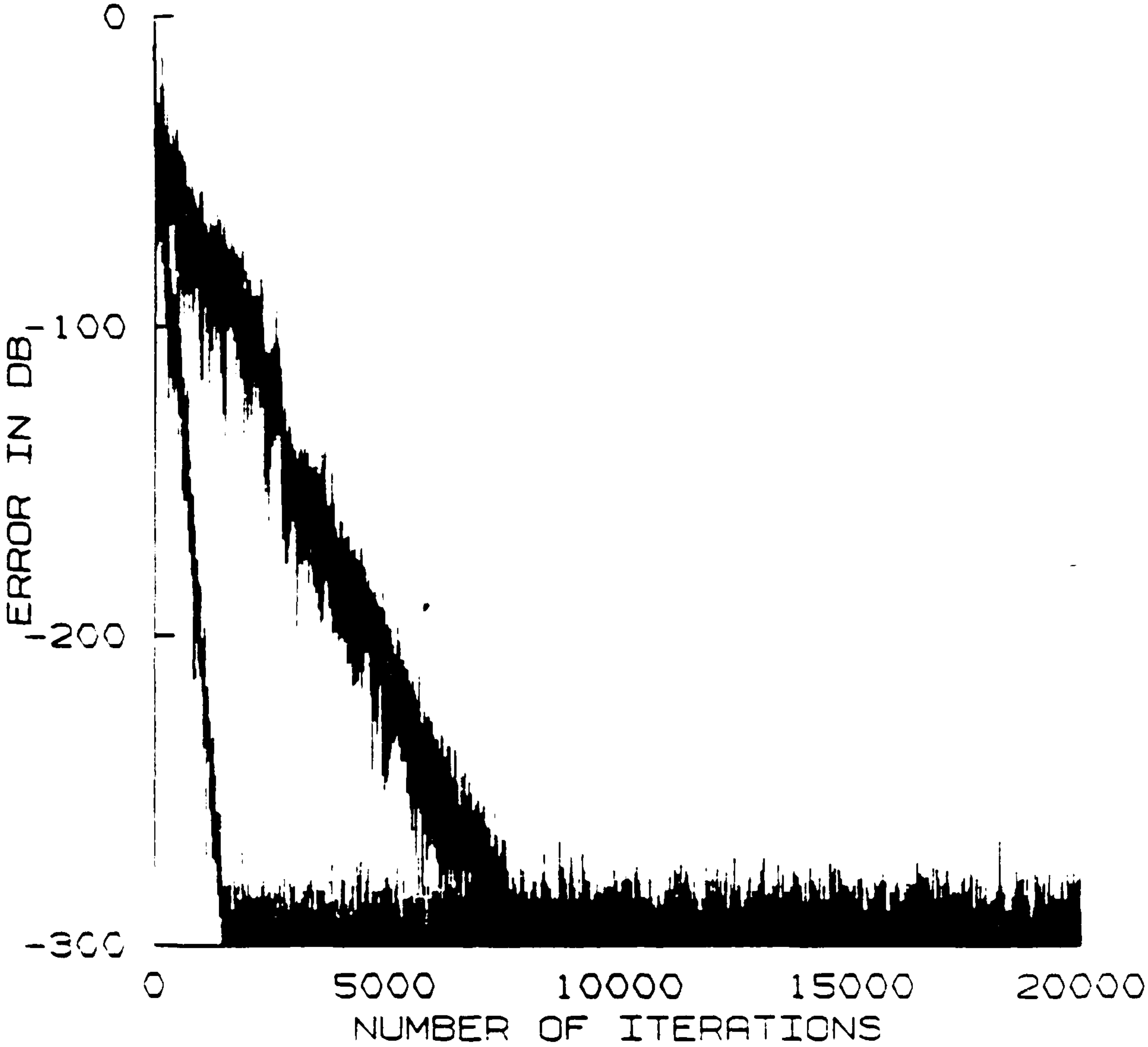


Figure 5.31 The PO2 versus time for band-reject noise

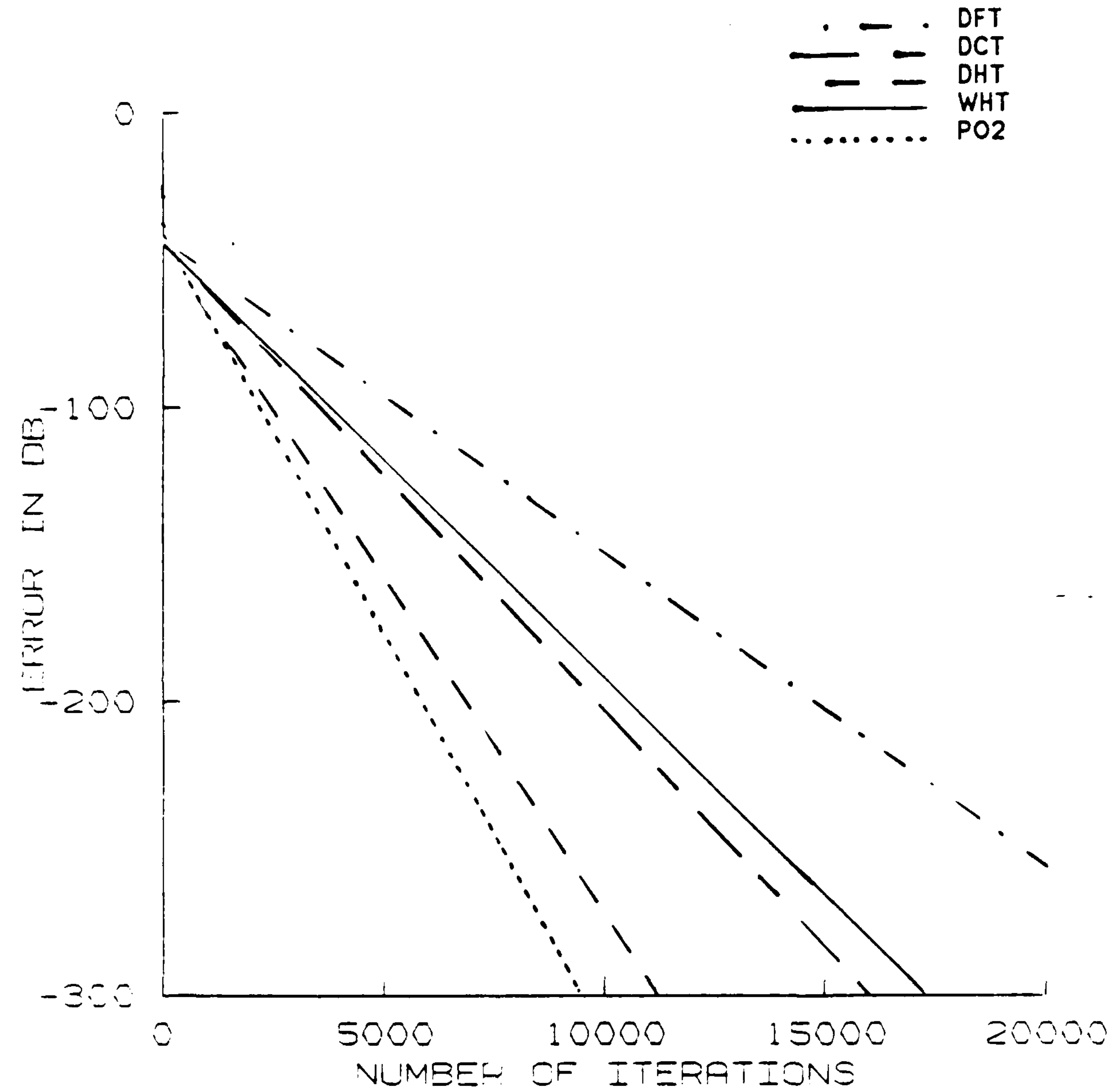


Figure 5.32 A comparison of transform performance for high-pass noise

5.4 A Comparison of Transforms

Once it has been established that the transform domain will result in faster convergence, it would be interesting to determine which transform will result in the greatest advantage over time for each given input, and which transform will provide the greatest advantage over the range of inputs taken as a whole. Another set of figures has been prepared that shows the learning curves for each transform superimposed on one axis for each colored noise input. Figure 5.32 shows the performance of the five transforms with high-pass noise. Because the original curves were noisy, smoothing was performed by hand. The smoothed curves are only to allow comparison of the transforms' relative performance. The actual learning curves are available compared against the untransformed adaptive filter. The smoothing was necessary to allow unobstructed viewing of each curve. For the high-pass noise, all the transforms except the DFT allowed convergence in under 20,000 iterations. The fastest convergence has been accomplished by the DHT and the PO2, with the PO2 seemingly finishing just slightly ahead.

A very different result is evident with the use of the low-pass model. As the set of curves in Figure 5.33 shows, only two of the transforms have allowed convergence within 20,000 iterations, and one, the DCT, has converged almost as fast as if the input were white noise. The only other transform to converge, the DHT, shows a precipitous drop in the first 100 or so adaptations, resulting in a curve much like the DCT for that span, but then the slope of the line changes, and

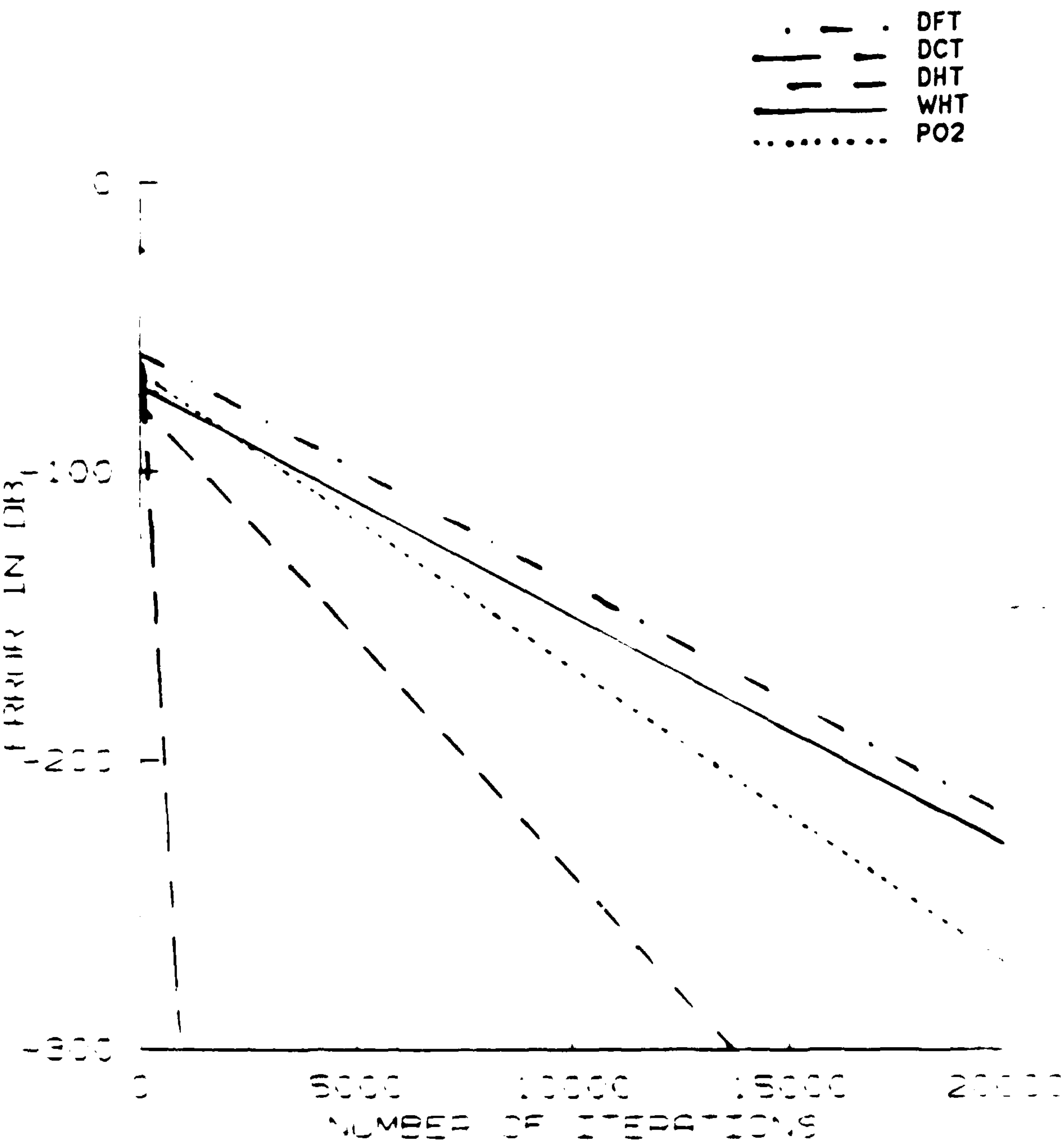


Figure 5.33 A comparison of transform performance for low-pass noise

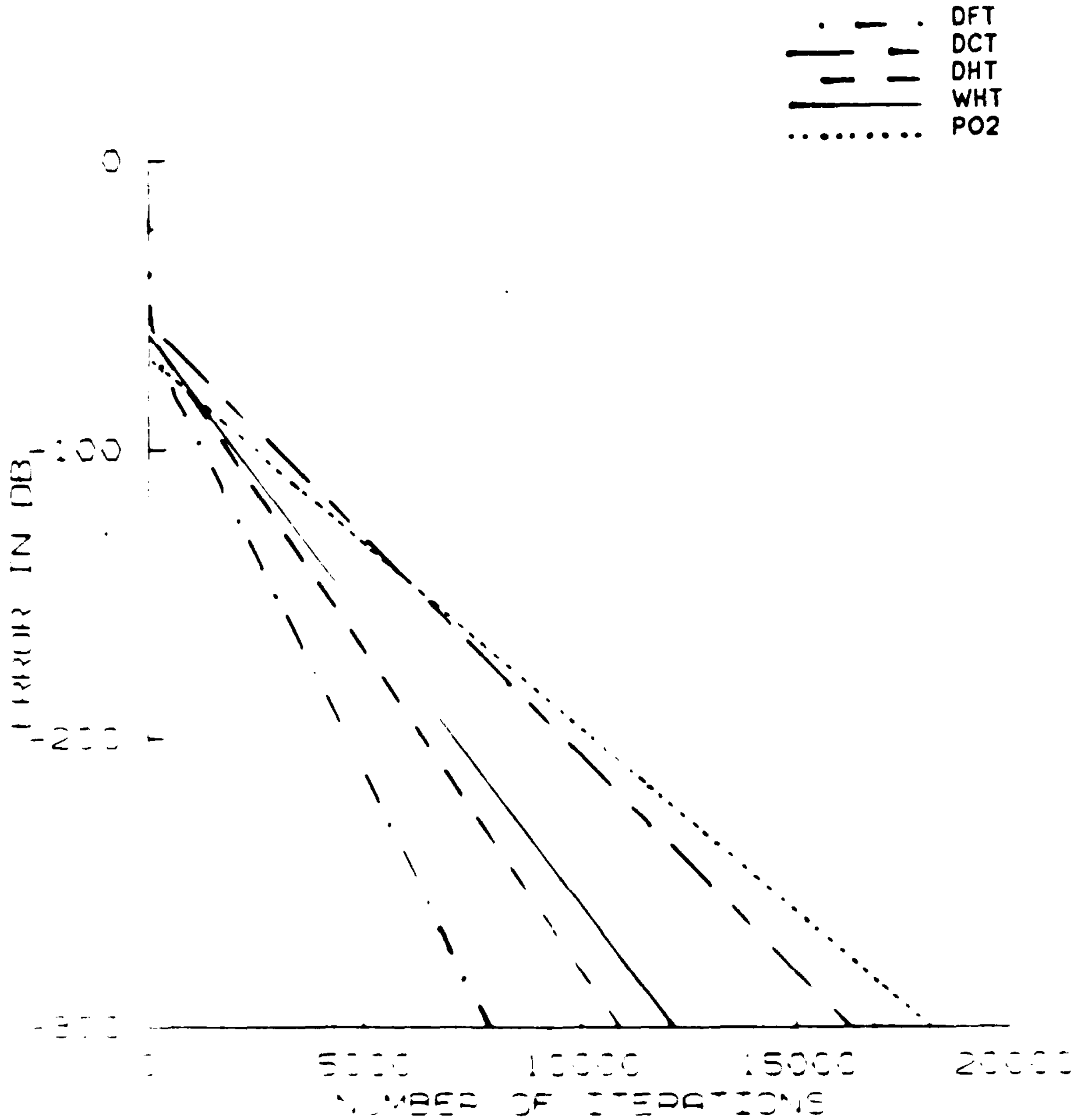


Figure 5.34 A comparison of transform performance for band-pass noise

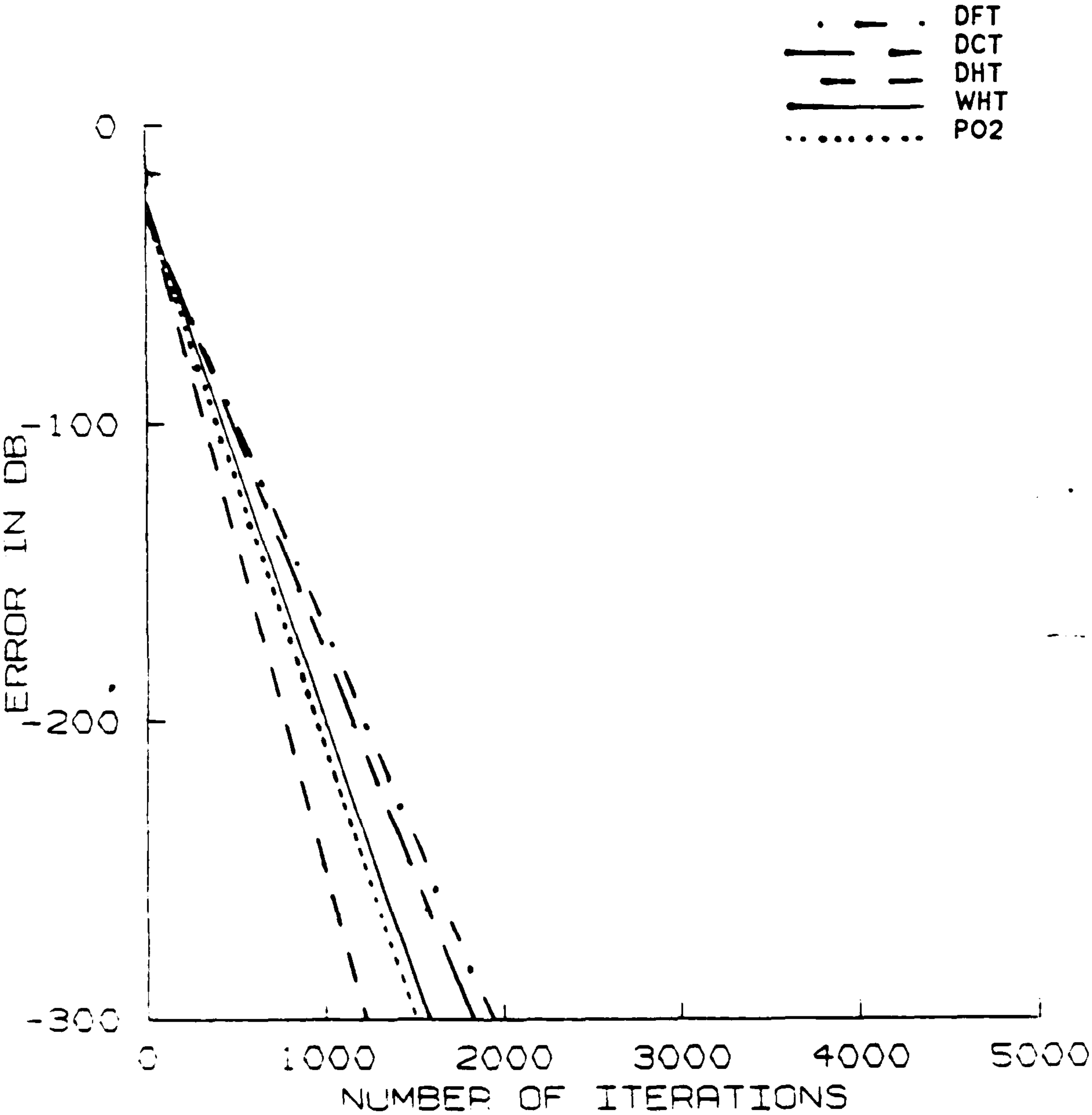


Figure 5.35 A comparison of transform performanmce for band-reject noise (5000 iterations)

at convergence the DHT has taken about 12,500 iterations to find the noise floor of the CYBER.

For the band-pass filter, all the transforms allow convergence in less than 20,000 iterations. On this run, the first transform to converge is the DFT, followed by the DHT. See Figure 5.34.

While the other three colorations resulted in system convergence time on the order of 20,000 iterations, the final filter, the band-reject case, has converged in the time domain in less than 10,000 iterations. When the transform is added, convergence occurs in less than 2000 iterations as a worst case. The fastest transform in this case appears to be the DHT, in about 1100 iterations, followed by the WHT. See Figure 5.35.

For any given input one transform will be optimal. Thus, for four different inputs, there was a different fastest transform for each. To derive a ranking, a transform was given a number corresponding to where it finished relative to the other transforms tested for each input. A number 1 was given to the transform that converged first, etc.. Table 5.1 shows the numbers given to all the transforms for the four inputs. In some cases, ties have been declared when the learning curves are very close. From the table, one can see that the most consistent transform by far is the DHT. It is the best in one case, and never falls to lower than second place on any input. While in certain cases other transforms may converge much more quickly, the consistency of the Hartley Transform makes it more versatile, allowing its use in many applications, including the one outlined in the introduction.

Table 5.1 A comparison of transforms

	High Pass	Low Pass	Band Pass	Band Reject	Rank
DFT	5	5	1	5	5
DCT	3	1	4	4	3
DHT	2	2	2	1	1
WHT	4	4	3	3	4
PO2	1	3	5	2	2

5.5 Computational Complexity

Computational complexity refers to the amount of calculation needed to implement any algorithm. Part of the attraction of the original LMS algorithm was the ease of computation brought about by the gradient estimate. One of the purposes of this research was to study ways to increase the convergence rate of the LMS algorithm, while keeping the level of complexity to that which could be accommodated by established VLSI architecture. The architecture presently in use may lag behind the state of the art, or may for some other reason be unable to accommodate a more complicated algorithm. Thus, the best algorithm can be rendered useless in some instances if the issue of its computation is ignored. Any good programmer can reduce the amount of computation necessary in any process to some extent, but a baseline will eventually be reached. In the case of the DFT, it is known that to calculate the FFT of an N point sequence requires on the order of $N\log_2N$ complex multiplies, and a like number of additions [1]. A complex multiply requires four real multiplies. Thus, the first factor to be examined is complex versus real arithmetic. There is no denying that the Fourier is a powerful transform. However, it has been shown that the Hartley Transform allows the calculation of the Fourier with only real arithmetic. Furthermore, when the transform rankings show the Hartley to be a more consistent performer, it becomes even more attractive. It has been shown [15, 16] that the DFT can be calculated by using an in-place Fast Hartley Transform algorithm about twice as

fast as a normal complex FFT. There are some more sophisticated FFT routines that can approach that speed, but the important thing is that we do not necessarily want the DFT. In most cases, our purposes are better served by the FHT, which is real, and about twice as fast as the FFT. The same savings are available to the other real transforms. The Walsh-Hadamard Transform can be calculated easily by using the FFT structure, making everything real, and setting all the exponential multipliers equal to one [20]. The PO2 Transform can also be calculated with no multiplications. Since all of its values are powers of two, the computer need only use shifts and adds to calculate the outputs [10]. Again, this will result in substantial computational savings over the FFT, but the performance of the WHT and PO2, as well as that of the DCT, is not as consistent as that of the DHT.

CHAPTER 6

CONCLUSIONS

6.1 Summary

A more complete investigation of various transforms used to improve the performance of the adaptive LMS algorithm was undertaken in order to provide a hierarchy of those transforms. The transforms studied, the DFT, DCT, DHT, WHT, and PO2, were all compared to the time-domain algorithm for four different common types of signalling conditions, to show that all did indeed provide some advantage with each of those types of inputs. The transforms were then compared to one another in order to establish which transform provided the best advantage with each input, and which one provided the best overall performance. In the last case, the DHT was shown to be the most consistent performer, though at a higher computational cost than either the WHT or PO2.

6.2 Future Directions

No transform performed well enough under all conditions to be declared the best transform to use with the LMS algorithm, though the Hartley performed well enough to recommend its use if transform-domain adaptive digital filtering is required. The difficulty in establishing a clear favorite experimentally, coupled with the failure to derive analytically a measure that would show conclusively which transform is the best overall, perhaps signals that this particular direction has gone as far as it will. Already attention has turned to other methods of improving the performance of adaptive filters. The transform LMS

algorithm still has worth in any system that needs a low-cost algorithm with better performance, and of the transforms tested in this work, the Hartley Transform is recommended as the best overall transform.

REFERENCES

[1] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1975.

[2] B. Widrow and S. Sterns, Adaptive Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1985.

[3] R. N. Bracewell, The Fourier Transform and Its Applications. New York: Mc Graw-Hill: 1986.

[4] R. N. Bracewell, The Hartley Transform. New York: Oxford University Press, 1986.

[5] R. N. Bracewell, "Discrete hartley transform," Journal of the Optical Society of America, vol. 73, no.12, December 1983.

[6] R. N. Bracewell, "The fast hartley transform," Proceedings of The IEEE, vol. 72, no. 8, August 1984.

[7] J. R. Kreidle, "Transform domain adaptive filtering," M.S. thesis, University of Illinois, Urbana Il, 1986.

[8] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform domain adaptive algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-31, no. 3, June 1983.

[9] W. K. Jenkins and D. F. Marshall, "On the analysis and design of unitary transformations for adaptive FIR digital filters," Proceedings of the 29th Midwest Symposium on Circuits and Systems, 1986.

[10] W. K. Jenkins, D. F. Marshall, J. R. Kreidle, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filters," IEEE Transactions on Acoustics, Speech, and Signal Processing, to be published.

[11] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Transactions on Computers, January 1974.

[12] R. V. Hartley, "A more symmetrical fourier applied to transmission problems," Proceedings of the Institute of Radio Engineers, vol. 30, 1942.

[13] D. F. Marshall, ADAPOP4: FORTRAN Program: 1986.

[14] E. J. Diethorn, OFDZR1: FORTRAN Program: 1984.

[15] J. Prado, "Comments on "the fast hartley transform"," Proceedings of

the IEEE, vol. 73, no. 12; December 1985.

[16] G. E. J. Bold, "A comparison of the time involved in computing fast hartley and fast fourier transforms," Proceedings of the IEEE, vol. 73, no. 12, December 1985.

[17] D. F. Marshall, The Transform Adaptive Filter: New Interpretations and Extensions, A dissertation proposal, University of Illinois, Urbana, Il, December 1986.

[18] M. Hamidi and J. Pearl, "Comparison of the cosine and fourier transforms of markov-1 signals," IEEE Transactions on Acoustics, Speech, and Signal Processing, October, 1976.

[19] P. Yip and D. Hutchinson, "Residual correlation for generalized discrete transforms," IEEE Transactions on Electromagnetic Compatibility, vol. EMC-24, no. 1, February, 1982.

[20] J. E. Whelchel and D. F. Guinn, "The fast fourier-hadamard transform and its use in signal representation and classification," EASCON 1968 Convention Record, 1968.

END
8-87
DTIC